

**IMPLEMENTASI SISTEM MONITORING PADA RUMAH  
JAMUR MENGGUNAKAN JARINGAN NIRKABEL BERBASIS  
PROTOKOL KOMUNIKASI MESSAGE QUEUING TELEMETRY  
TRANSPORT (MQTT)**

**SKRIPSI**

**KEMINATAN TEKNIK KOMPUTER**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Nugraha Pangestu  
NIM: 115060900111005



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018**

## PENGESAHAN

IMPLEMENTASI SISTEM MONITORING PADA RUMAH JAMUR MENGGUNAKAN  
JARINGAN NIRKABEL BERBASIS PROTOKOL KOMUNIKASI MESSAGE QUEUING  
TELEMETRY TRANSPORT (MQTT)

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

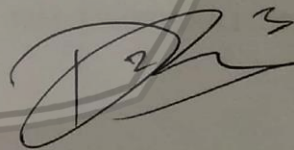
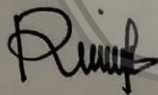
Disusun Oleh :  
Nugraha Pangestu  
NIM: 115060907111005

Skripsi ini telah diuji dan dinyatakan lulus pada  
02 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing 1

Dosen Pembimbing 2



Rizal Maulana, S.T, M.T., M.Sc.  
NIK. 2016078910091001

Rakhmadhany Pramananda, S.T., M.Kom  
NIK. 2016098604061001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurmiawan, S.T, M.T, Ph.D.  
NIP. 19710518 200312 1 001

A

## IDENTITAS PENGUJI

**Penguji 1 : Ari Kusyanti, S.T, M.Sc**

NIK : 201102 831228 2 001

**Penguji 2 : Ir. Primantara Hari Trisnawan, M. Sc.**

NIK : 19680912 199403 1 002





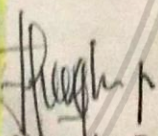
## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 14 Juli 2018



  
Nugraha Pangestu

NIM: 115060900111005

## DAFTAR RIWAYAT HIDUP

### Biodata

Nama : Nugraha Pangestu

Alamat: Jambu, Kepuharjo, Cangkringan, Sleman, Yogyakarta

No. HP : 085785600009

Email : [nugraha.pangestu@gmail.com](mailto:nugraha.pangestu@gmail.com)

### Pendidikan

<i>Sekolah</i>	<i>Tahun Lulus</i>
Universitas Brawijaya, Teknik Informatika	
SMK N 3 Yogyakarta	2011
MTs N Pakem	2008
SD N Petung	2005

## UCAPAN TERIMA KASIH

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Kedua orang tua dan adik tercinta yang selalu memberikan dukungan doa serta motivasi sehingga penulis dapat menyelesaikan skripsi ini,
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D sebagai dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang,
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D sebagai Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang,
4. Bapak Sabriansyah Rizqika Akbar, S.T, M. Eng sebagai Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang yang selalu mendorong penulis agar menyelesaikan penulisan skripsi ini,
5. Bapak Adharul Muttaqin, S.T, M.T sebagai Dosen Penasihat Akademik sejak Semester I yang selalu memberikan semangat untuk menyelesaikan penulisan skripsi yang tertunda selama ini,
6. Bapak Rizal Mulana, S.T, M.T., M.Sc. selaku dosen pembimbing 1 yang selalu membimbing dan mengarahkan penulis agar dapat menyelesaikan skripsi ini dengan baik dan benar sesuai konteks kaidah penelitian,
7. Bapak Rakhmadhany Primananda, S.T,. M.Kom selaku dosen pembimbing 2 yang selalu sabar dalam membimbing dan mengarahkan penulis dalam menyelesaikan skripsi ini,
8. Kepada teman-teman Siskom 2011, Emirza Alam, Anake Deborah, Yuni Mustika, Reza Akhmad, Alfaviaga, Ryan Adhisa, Vynska, Kholil Gibran, Bagus Priyo, M. Nur Arifin, Loki Sudiarta, Riskita, Rahadian Sayogo, Sunu Dias, dan seluruh teman-teman Siskom 2011 yang sudah memberikan semangat dan motivasi untuk penyelesaian skripsi ini,

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun akan sangat diharapkan penulis. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

## ABSTRAK

Budidaya jamur pada saat ini terus berkembang pada lingkungan masyarakat. Jamur menjadi bahan pangan yang digemari masyarakat karena harganya yang cukup terjangkau untuk semua kalangan masyarakat. Suhu dan kelembaban ruangan menjadi faktor jamur dapat tumbuh dengan baik sebab jika suhu dan kelembaban tidak sesuai dengan batas ideal maka tanaman jamur tidak dapat tumbuh dengan baik. Suhu ideal untuk pertumbuhan jamur berkisar diantara 25-30°C dan kelembaban udara berkisar diantara 60-90%. Untuk mengatasi permasalahan tersebut, dirancang sistem monitoring suhu dan kelembaban pada rumah jamur menggunakan protokol MQTT. Untuk mendeteksi suhu dan kelembaban digunakan sensor DHT-22, data dari sensor akan diproses oleh mikrokontroler Arduino dan dikirimkan menggunakan modul *Wi-fi* ESP8266-01 yang terhubung ke jaringan *Wi-fi* ke server Thingsboard menggunakan protokol MQTT dan data akan ditampilkan dalam bentuk chart dan grafik. Dalam sistem ini dilakukan pengujian delay dengan menghitung waktu yang dibutuhkan sistem dari proses sensor membaca data hingga data dikirim dan ditampilkan pada server Thingsboard. Dari hasil pengujian delay yang dilakukan, sistem membutuhkan waktu rata-rata 1.20 detik dalam satu kali pengiriman data ke server Thingsboard.

Kata kunci : Monitoring, Suhu dan Kelembaban, Rumah Jamur, Thingsboard.



## ABSTRACT

Nowadays, mushroom cultivations are growing rapidly in the society. Mushrooms have been becoming the popular groceries because of its affordable price for all people. In its growth process, Mushroom will need two main factors in order to grow properly; which are : temperature and humidity of the room. These two factors need to be controlled continuously to fit the ideal value limit. The ideal temperature for mushroom growth ranges between 25-30 °C and the air humidity ranges between 60-90%. To overcome these problems, we proposed a system that can continuously monitor temperature and humidity of mushroom house using MQTT protocol. To detect temperature and humidity, DHT-22 sensor is used. The sensor data will be processed by Arduino microcontroller and then sent using ESP8266-01 Wi-fi module which connected through the Wi-fi network to the Thingsboard server using MQTT protocol. Finally the data will be displayed in the form of charts and graphs. We evaluate the system using the delay test by calculating the time required by each sensor to process and read the data until the data is sent and displayed on the Thingsboard server. The results of delay test have confirmed that the system takes an average of 1.20 seconds in a single delivery of data to the Thingsboard server.

**Keywords:** Monitoring, Temperature and Humidity, Mushroom House, Thingsboard.



## KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Sistem Monitoring Pada Rumah Jamur Menggunakan Jaringan Nirkabel Berbasis Protokol Komunikasi Message Queuing Telemetry Transport (MQTT)” ini dapat terselesaikan.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Kedua orang tua dan adik tercinta yang selalu memberikan dukungan doa serta motivasi sehingga penulis dapat menyelesaikan skripsi ini,
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D sebagai dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang,
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D sebagai Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang,
4. Bapak Sabriansyah Rizqika Akbar, S.T, M. Eng sebagai Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang yang selalu mendorong penulis agar menyelesaikan penulisan skripsi ini,
5. Bapak Adharul Muttaqin, S.T, M.T sebagai Dosen Penasihat Akademik sejak Semester I yang selalu memberikan semangat untuk menyelesaikan penulisan skripsi yang tertunda selama ini,
6. Bapak Rizal Mulana, S.T, M.T., M.Sc. selaku dosen pembimbing 1 yang selalu membimbing dan mengarahkan penulis agar dapat menyelesaikan skripsi ini dengan baik dan benar sesuai konteks kaidah penelitian,
7. Bapak Rakhmadhany Primananda, S.T., M.Kom selaku dosen pembimbing 2 yang selalu sabar dalam membimbing dan mengarahkan penulis dalam menyelesaikan skripsi ini,
8. Segenap Bapak dan Ibu dosen Fakultas Ilmu Komputer Universitas Brawijaya atas segenap ilmu pengetahuan dan perhatian yang diberikan,
9. Kepada teman-teman Siskom 2011, Emirza Alam, Anake Deborah, Yuni Mustika, Reza Akhmad, Alfaviaga, Ryan Adhisa, Vynska, Kholil Gibran, Bagus Priyo, M. Nur Arifin, Loki Sudiarta, Riskita, Rahadian Sayogo, Sunu Dias, dan seluruh teman-teman Siskom 2011 yang sudah memberikan semangat dan motivasi untuk penyelesaian skripsi ini,

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun akan sangat diharapkan

penulis. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 23 Maret 2018

Penulis

Nugraha Pangestu



## DAFTAR ISI

PENGESAHAN .....	<b>Error! Bookmark not defined.</b>
PERNYATAAN ORISINALITAS .....	ii
KATA PENGANTAR.....	iv
ABSTRAK.....	vii
ABSTRACT .....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN .....	xvi
BAB 1 PENDAHULUAN.....	<b>Error! Bookmark not defined.</b>
1.1 Latar belakang.....	<b>Error! Bookmark not defined.</b>
1.2 Rumusan masalah.....	<b>Error! Bookmark not defined.</b>
1.3 Tujuan .....	<b>Error! Bookmark not defined.</b>
1.4 Manfaat.....	<b>Error! Bookmark not defined.</b>
1.5 Batasan masalah .....	<b>Error! Bookmark not defined.</b>
1.6 Sistematika Pembahasan.....	<b>Error! Bookmark not defined.</b>
BAB 2 LANDASAN KEPUSTAKAAN .....	<b>Error! Bookmark not defined.</b>
2.1 Kajian Pustaka .....	<b>Error! Bookmark not defined.</b>
2.2 Dasar Teori.....	<b>Error! Bookmark not defined.</b>
2.2.1 Jamur Tiram.....	<b>Error! Bookmark not defined.</b>
2.2.2 Protokol MQTT .....	<b>Error! Bookmark not defined.</b>
2.2.3 Website .....	<b>Error! Bookmark not defined.</b>
2.2.4 THINGSBOARD.....	<b>Error! Bookmark not defined.</b>
2.2.5 ARDUINO UNO .....	<b>Error! Bookmark not defined.</b>
2.2.6 DHT – 22 .....	<b>Error! Bookmark not defined.</b>
2.2.7 Modul ESP8266-01 .....	<b>Error! Bookmark not defined.</b>
BAB 3 METODOLOGI PENELITIAN .....	<b>Error! Bookmark not defined.</b>
3.1 Studi Literatur .....	<b>Error! Bookmark not defined.</b>
3.2 Analisis Kebutuhan .....	<b>Error! Bookmark not defined.</b>
3.3 Perancangan Sistem.....	<b>Error! Bookmark not defined.</b>
3.4 Implementasi Sistem .....	<b>Error! Bookmark not defined.</b>

3.5 Pengujian .....	<b>Error! Bookmark not defined.</b>
BAB 4 REKAYASA KEBUTUHAN .....	<b>Error! Bookmark not defined.</b>
4.1 Deskripsi Umum.....	<b>Error! Bookmark not defined.</b>
BAB 5 PERANCANGAN DAN IMPLEMENTASI .....	<b>Error! Bookmark not defined.</b>
5.1 Perancangan Perangkat Keras .....	<b>Error! Bookmark not defined.</b>
5.1.1 Perancangan Koneksi Sensor .....	<b>Error! Bookmark not defined.</b>
5.1.2 Perancangan Koneksi Modul ESP8266-01.....	<b>Error! Bookmark not defined.</b>
5.1.3 Perancangan Pengiriman Data Menggunakan MQTT .....	<b>Error! Bookmark not defined.</b>
5.2 Perancangan Perangkat Lunak .....	<b>Error! Bookmark not defined.</b>
5.2.1 Pembacaan Data Pada Sensor .....	<b>Error! Bookmark not defined.</b>
5.2.2 Pengiriman Data.....	<b>Error! Bookmark not defined.</b>
5.3 Implementasi Sistem .....	<b>Error! Bookmark not defined.</b>
5.3.1 Implementasi Hardware.....	<b>Error! Bookmark not defined.</b>
5.3.2 Implementasi Software .....	<b>Error! Bookmark not defined.</b>
5.3.3 Implementasi Sistem Pemberitahuan Suhu dan Kelembaban .....	<b>Error! Bookmark not defined.</b>
5.3.4 Hasil Pengiriman Data ke Web Server.....	<b>Error! Bookmark not defined.</b>
BAB 6 PENGUJIAN DAN analisis .....	<b>Error! Bookmark not defined.</b>
6.1 Pengujian .....	<b>Error! Bookmark not defined.</b>
6.1.1 Pengujian Pembacaan Data Sensor .....	<b>Error! Bookmark not defined.</b>
6.1.2 Pengujian Delay pada MQTT .....	<b>Error! Bookmark not defined.</b>
6.1.3 Pengujian Fungsionalitas Sistem .....	<b>Error! Bookmark not defined.</b>
6.2 Analisis .....	<b>Error! Bookmark not defined.</b>
BAB 7 KESIMPULAN DAN SARAN .....	<b>Error! Bookmark not defined.</b>
7.1 Kesimpulan.....	<b>Error! Bookmark not defined.</b>
7.2 Saran .....	<b>Error! Bookmark not defined.</b>
Daftar Pustaka .....	<b>Error! Bookmark not defined.</b>
LAMPIRAN A KODE PROGRAM.....	<b>Error! Bookmark not defined.</b>
LAMPIRAN B GAMBAR ALAT DAN THINGSBOARD .....	<b>Error! Bookmark not defined.</b>
LAMPIRAN C PERBANDINGAN SENSOR.....	<b>Error! Bookmark not defined.</b>



## DAFTAR GAMBAR

Gambar 2.1 Mekanisme pengiriman data protokol MQTT	Error! Bookmark not defined.
Gambar 2.2 Arduino Uno .....	Error! Bookmark not defined.
Gambar 2.3 Sensor DHT-22.....	Error! Bookmark not defined.
Gambar 2.4. ESP8266-01.....	Error! Bookmark not defined.
Gambar 3.1 Alur Pelaksanaan Penelitian .....	Error! Bookmark not defined.
Gambar 3.2 Diagram Blok Sistem .....	Error! Bookmark not defined.
Gambar 4.1 Diagram Kebutuhan Perangkat Keras Sistem	Error! Bookmark not defined.
Gambar 5.1 Perancangan umum sistem.....	Error! Bookmark not defined.
Gambar 5.2 Sensor DHT-22.....	Error! Bookmark not defined.
Gambar 5.3 Koneksi sensor DHT-22 ke Arduino.....	Error! Bookmark not defined.
Gambar 5.4 Modul ESP8266-01 .....	Error! Bookmark not defined.
Gambar 5.5 Perancangan MQTT .....	Error! Bookmark not defined.
Gambar 5.6 Flowchart Sistem Monitoring Suhu dan Kelembaban	Error! Bookmark not defined.
Gambar 5.7 Skema Rangkain Sensor Suhu dan Kelembaban	Error! Bookmark not defined.
Gambar 5.8 Rangkaian Sensor Suhu dan Kelembaban	Error! Bookmark not defined.
Gambar 5.9 Server Attribute Thingsboard.....	Error! Bookmark not defined.
Gambar 5.10 Message Telemetry Filter .....	Error! Bookmark not defined.
Gambar 5.11 Menentukan ZoneType Filter pada Thingsboard	Error! Bookmark not defined.
Gambar 5.12 Menentukan Suhu dan Kelembaban pada Sistem	Error! Bookmark not defined.
Gambar 5.13 Alarm Prosesor pada Sistem .....	Error! Bookmark not defined.
Gambar 5.14 Konfigurasi Email Plugin .....	Error! Bookmark not defined.
Gambar 5.15 Tampilan Devices pada Server Thingsboard	Error! Bookmark not defined.
Gambar 5.16 Akses Token Devices .....	Error! Bookmark not defined.
Gambar 5.17 Tampilan Data Menggunakan Grafik dan Chart	Error! Bookmark not defined.

Gambar 5.18 Email Pemberitahuan Suhu dan Kelembaban**Error! Bookmark not defined.**

Gambar 6.1 Data Sensor sebelum didekatkan pada nyala api**Error! Bookmark not defined.**

Gambar 6.2 Data Sensor sesudah didekatkan pada nyala api**Error! Bookmark not defined.**

Gambar 6.3 Suhu dan kelembaban pada termometer**Error! Bookmark not defined.**

Gambar 6.4 Suhu dan kelembaban dari sensor DHT-22**Error! Bookmark not defined.**

Gambar 6.5 Kondisi rumah jamur .....**Error! Bookmark not defined.**

Gambar 6.6 Hardware sistem monitoring suhu dan kelembaban**Error! Bookmark not defined.**

Gambar 6.7 Tampilan data suhu dan kelembaban pada Thingsboard .....**Error! Bookmark not defined.**



## DAFTAR TABEL

Tabel 2-1 Kajian Pustaka .....	<b>Error! Bookmark not defined.</b>
Tabel 2-2 Pin Modul ESP8266-01 .....	<b>Error! Bookmark not defined.</b>
Tabel 5-1 Kode Sumber Library Arduino .....	<b>Error! Bookmark not defined.</b>
Tabel 5-2 Kode Sumber Inisialisasi Access Point dan Token Thingsboard .....	<b>Error! Bookmark not defined.</b>
Tabel 5-3 Kode Sumber Definisi Pin dan Tipe Sensor DHT-22	<b>Error! Bookmark not defined.</b>
Tabel 5-4 Kode Sumber Definisi alamat server Thingsboard	<b>Error! Bookmark not defined.</b>
Tabel 5-5 Kode Sumber Inisialisasi fungsi pada program	<b>Error! Bookmark not defined.</b>
Tabel 5-6 Kode Sumber Inisialisasi Modul ESP8266-01 ke Arduino .....	<b>Error! Bookmark not defined.</b>
Tabel 5-7 Kode Sumber Koneksi Modul ESP8266-01 ke jaringan Wi-fi .....	<b>Error! Bookmark not defined.</b>
Tabel 5-8 Kode Sumber Penghitungan waktu proses pada program .....	<b>Error! Bookmark not defined.</b>
Tabel 5-9 Kode Sumber Pengambilan nilai data sensor	<b>Error! Bookmark not defined.</b>
Tabel 5-10 Kode Sumber Payload pengiriman paket data	<b>Error! Bookmark not defined.</b>
Tabel 5-11 Kode Sumber Mengirim payload ke Thingsboard	<b>Error! Bookmark not defined.</b>
Tabel 6-1 Hasil Pengujian Waktu Ekekusi Program pada Arduino	<b>Error! Bookmark not defined.</b>
Tabel 6-2 Hasil Pengujian Waktu Data yang Tampil di Thingsboard .....	<b>Error! Bookmark not defined.</b>
Tabel 6-3 Penghitungan delay pengiriman data dari Thingsboard dikurangi waktu pada serial monitor .....	<b>Error! Bookmark not defined.</b>

## DAFTAR LAMPIRAN

LAMPIRAN A KODE PROGRAM .....	50
LAMPIRAN B GAMBAR ALAT DAN THINGSBOARD.....	54
LAMPIRAN C PERBANDINGAN SENSOR .....	57







## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Budidaya jamur pada saat ini terus berkembang pada lingkungan masyarakat. Jamur menjadi bahan pangan yang digemari masyarakat karena harganya yang cukup terjangkau untuk kalangan masyarakat. Banyak restoran-restoran ataupun warung makan menyediakan menu spesial jamur. Menurut data Badan Statistik Daerah Istimewa Yogyakarta (BPS DIY) pada tahun 2016 dapat diketahui bahwa tanaman jamur merupakan produk hasil pertanian dengan jumlah tertinggi dibandingkan dengan komoditas lainnya yaitu dengan jumlah 1.349.305 kuintal, hal ini menunjukkan bahwa jamur merupakan produk hasil pertanian yang paling banyak digemari dan dikembangkan oleh masyarakat.

Terdapat berbagai macam jenis jamur yang dapat dibudidayakan. Jenis-jenis jamur yang umum dibudidayakan adalah jamur merang, jamur tiram, jamur kuping dan jamur payung. Jamur tiram merupakan jenis jamur yang paling banyak dibudidayakan oleh petani jamur karena banyaknya permintaan dari konsumen. Kandungan nutrisi jamur tiram lebih baik dibandingkan dengan jenis jamur lainnya. Jamur tiram mempunyai kandungan nilai gizi setiap 100 gram jamur kering juga mengandung protein 10,5 - 30,4%, lemak 1,7 - 2,2%, karbohidrat 56,6%, tiamin 0,2 mg, riboflavin 4,7 - 4,9 mg, niasin 77,2 mg, kalsium 314 mg, dan kalori 367. (Suwito, M, 2006). Budidaya jamur tiram sangat mudah dilakukan dan tidak mengenal musim. Jamur dapat tumbuh dan berkembang dengan baik dengan dipengaruhi oleh penyiapan ruangan sebagai tempat untuk tumbuh jamur. Suhu dan kelembaban ruangan menjadi faktor jamur dapat tumbuh dengan baik sebab jika suhu dan kelembaban tidak sesuai dengan batas ideal yang diperlukan maka jamur dapat mengering dan tanaman jamur tidak dapat tumbuh dengan baik. Jika suhu pada ruangan dibawah suhu ideal mengakibatkan tubuh buahnya mengecil dan tangkainya panjang tapi kurus dan apabila suhu pada ruangan diatas suhu ideal maka akan menyebabkan payung jamur menjadi tipis dan ukurannya kerdil. Kelembaban ruangan yang terlalu tinggi juga menyebabkan tubuh buah jamur cepat membusuk dan kelembaban ruang yang terlalu rendah menyebabkan tubuh jamur menjadi kurus. Suhu ideal untuk tumbuh kembang jamur berkisar diantara 25-30°C dan kelembaban udara berkisar diantara 60-90%. (Djarjah, 2001)

Berdasarkan latar belakang tersebut untuk memudahkan petani dalam melakukan budidaya jamur tiram maka dibutuhkan sistem monitoring untuk memantau suhu dan kelembaban pada rumah jamur menggunakan protokol MQTT. Pada protokol MQTT terdiri dari *publisher* dan *subscriber*. *Publisher* merupakan perangkat yang mengirimkan data menuju ke server dan ke

*subscriber*. Sedangkan *subscriber* merupakan perangkat penerima pesan dari server maupun *publisher*. Dalam proses komunikasi antara *publisher* dan *subscriber* diperlukan adanya *broker* yang bertanggung jawab untuk mengatur pengiriman pesan. Pada sistem ini *broker* menggunakan *platform Thingsboard* dan data akan ditampilkan dalam bentuk *chart* dan grafik.

## 1.2 Rumusan masalah

1. Bagaimana merancang sistem monitoring pada rumah jamur menggunakan jaringan nirkabel berbasis protokol *MQTT*
2. Bagaimana mengimplementasikan sistem monitoring pada rumah jamur menggunakan jaringan nirkabel berbasis protokol *MQTT*
3. Bagaimana hasil yang diperoleh pada sistem monitoring pada rumah jamur menggunakan jaringan nirkabel berbasis protokol *MQTT*

## 1.3 Tujuan

1. Merancang sistem monitoring pada rumah jamur menggunakan jaringan sistem nirkabel berbasis protokol *MQTT*
2. Mengimplementasikan sistem monitoring pada rumah jamur menggunakan jaringan sistem nirkabel berbasis protokol *MQTT*
3. Mendapatkan hasil optimal dari implementasi sistem monitoring pada rumah jamur menggunakan jaringan sistem nirkabel berbasis protokol *MQTT*

## 1.4 Manfaat

1. Untuk dapat mengetahui perubahan nilai suhu dan kelembaban yang ada pada rumah jamur
2. Untuk dapat menentukan langkah-langkah yang diperlukan serta menjaga kondisi suhu dan kelembaban pada lingkungan rumah jamur tetap stabil
3. Untuk dapat mengoptimalkan jumlah hasil produksi jamur dan meningkatkan keuntungan petani jamur.

## 1.5 Batasan masalah

Batasan masalah pada sistem adalah :

1. Mikrokontroller yang digunakan adalah Arduino UNO
2. Modul pengiriman data secara nirkabel menggunakan modul ESP8266.
3. Perangkat keras dari sistem dalam bentuk *prototype*
4. Jaringan yang digunakan untuk simulasi *MQTT* dan broker adalah menggunakan jaringan lokal

5. *Broker* pada sistem menggunakan *broker mosquitto*
6. *Web server* menggunakan platform *thingsboard*.

## 1.6 Sistematika Pembahasan

Penulisan laporan penelitian pada sistem ini terdiri dari beberapa bab sebagai berikut :

### **BAB I : PENDAHULUAN**

Pada Bab ini terdiri dari latar belakang masalah pada penelitian, tujuan, dan manfaat pembuatan penelitian, permasalahan, batasan masalah, dan sistematika penulisan laporan penelitian.

### **BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI**

Pada Bab ini berisi tentang kajian pustaka yang berisi tentang penelitian yang sudah dibuat sebelumnya oleh Joko Nugroho dan Totok Budioko untuk digunakan sebagai acuan penelitian yang akan dibuat oleh penulis serta terdapat dasar teori pendukung dalam pembuatan laporan penelitian.

### **BAB III : METODOLOGI PENELITIAN**

Pada bab ini berisi tentang metode – metode yang digunakan dan alur kerja penelitian

### **BAB IV : REKAYASA KEBUTUHAN**

Pada Bab menjelaskan tentang rekayasa kebutuhan sistem yang akan dibuat. Bab ini merupakan penjabaran dari metodologi penelitian yang ada pada Bab sebelumnya

### **BAB V : PERANCANGAN DAN IMPLEMENTASI**

Pada Bab ini membahas perancangan sistem meliputi perancangan perangkat lunak dan perangkat keras dari sistem dan implementasi sistem meliputi implementasi perangkat lunak yaitu penulisan kode program dan implementasi perangkat keras dari sistem.

### **BAB VI : PENGUJIAN**

Pada Bab ini menjelaskan tentang langkah-langkah pengujian sistem dan memberikan analisis serta menjabarkan hasil dari beberapa pengujian yang telah dilakukan

### **BAB VI : KESIMPULAN DAN SARAN**

Pada Bab ini menjabarkan kesimpulan dari rumusan masalah penelitian termasuk hasil dari pengujian penelitian yang sudah dilakukan serta memberikan saran untuk pengembang sistem yang lebih baik.



## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Pada penulisan penelitian ini dibutuhkan kajian pustaka untuk membandingkan penelitian sudah dilakukan sebelumnya dengan penelitian yang akan dibuat sebagai berikut :

**Tabel 2-1 Kajian Pustaka**

No	Nama Penulis, Tahun dan Judul	Persamaan	Perbedaan	
			Penelitian terdahulu	Rencana Penelitian
1	Joko, Nugroho [2014] Aplikasi Sistem Monitoring Pendeteksi Suhu dan Kelembaban Pada Rumah Jamur Berbasis Mikrokontroller AT-Mega 328	Sistem monitoring suhu dan kelembaban	Menggunakan sensor DHT-11 dan AT-Mega 328, data ditampilkan pada LCD	Menggunakan sensor DHT-22 dan mikrokontroller Arduino Uno, data ditampilkan pada <i>website</i> berupa <i>chart</i> dan grafik
2	Budioko, Totok [2016] Sistem Monitoring Suhu Jarak Jauh Berbasis Internet Of Things Menggunakan Protokol MQTT	Menggunakan Protokol MQTT untuk metode <i>transport</i> data	Sensor suhu menggunakan LM35, data suhu ditampilkan pada <i>website</i> dalam bentuk angka	Sensor suhu menggunakan DHT-22, data ditampilkan pada <i>website</i> dalam bentuk <i>chart</i> dan grafik

Pada Tabel 2-1 merupakan tabel penelitian sebelumnya yang dilakukan oleh Joko Nugroho pada tahun 2014 menjelaskan tentang pengawasan suhu dan kelembaban yang ada pada rumah jamur dengan basis mikrokontroller AT-Mega328 dan menggunakan sensor suhu dan kelembaban menggunakan sensor DHT-11. Cara kerja dari sistem tersebut adalah sensor DHT-11 membaca data suhu dan kelembaban pada rumah jamur dan selanjutnya data akan diproses oleh mikrokontroller AT-Mega 328 dan data dari pembacaan suhu dan kelembaban akan ditampilkan melalui layar LCD 16x2. (Joko Nugroho, 2014)

Pada penelitian selanjutnya yang dilakukan oleh Totok Budioko pada tahun 2016 yaitu membuat sistem monitoring suhu jarak jauh menggunakan protokol MQTT. Pada sistem tersebut untuk mendeteksi suhu menggunakan sensor LM35 dan data diproses oleh mikrokontroller Arduino. Untuk pengiriman data menggunakan jaringan *Wi-fi* melalui modul ESP8266. Data suhu pada sistem ditampilkan melalui *website*. (Budioko, 2016)

Oleh karena itu penulis melakukan penelitian untuk membuat sistem monitoring pada rumah jamur menggunakan jaringan nirkabel berbasis protokol MQTT. Sensor yang digunakan untuk mendeteksi suhu dan kelembaban adalah sensor DHT-22. Penggunaan sensor DHT-22 dikarenakan mempunyai akurasi pembacaan data lebih baik dengan tingkat error sebesar  $\pm 2\%$  dibandingkan dengan sensor DHT-11 yang memiliki tingkat error sebesar  $\pm 5\%$  dan sensor LM35 dengan tingkat error sebesar  $\pm 5\%$ . Untuk memudahkan pengguna sistem monitoring, hasil pemantauan suhu dan kelembaban akan ditampilkan pada *website* dalam bentuk *chart* dan grafik dengan tujuan agar sistem lebih mudah untuk diakses dibandingkan dengan menggunakan LCD 6x4. Mikrokontroler Arduino uno digunakan untuk melakukan pengolahan data pada sistem karena mempunyai banyak *library* program yang siap digunakan sesuai dengan kebutuhan sistem. Untuk pengiriman data pada sistem ini menggunakan modul *wi-fi* ESP8266-01 yang dapat terhubung dengan jaringan nirkabel.

## 2.2 Dasar Teori

### 2.2.1 Jamur Tiram

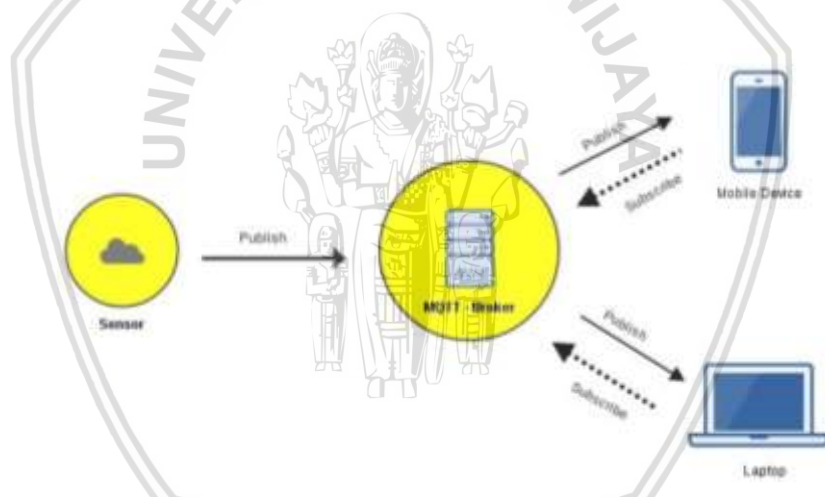
Jamur tiram dalam bahasa latin disebut dengan *Pleurotus ostreatus* yaitu jamur yang termasuk pada kelompok *Basidiomycota*. Jamur tersebut mempunyai ciri fisik berwarna putih dan berbentuk cekung seperti cangkang tiram. Jamur tiram biasanya tumbuh pada permukaan batang pohon yang sudah lapuk. Budidaya jamur ini dapat menggunakan media tanam seperti serbuk gergaji. (Chazali, Syammahfuz, 2010). Kandungan nutrisi yang terdapat pada jamur tiram yaitu setiap 100 gram jamur kering mengandung protein 10,5 - 30,4%, lemak 1,7 - 2,2%, karbohidrat 56,6%, tiamin 0,2 mg, riboflavin 4,7 - 4,9 mg, niasin 77,2 mg, kalsium 314 mg, dan kalori 367. (Suwito, M, 2006)

Pertumbuhan jamur dipengaruhi oleh beberapa faktor baik secara biotik maupun secara abiotik. Komponen biotik adalah komponen yang menyusun ekosistem, terdiri dari organisme yang ada didalamnya. Sedangkan komponen abiotik terdiri dari faktor fisik dan faktor kimia dari sebuah media tanam jamur. Faktor abiotik pada jamur tiram meliputi suhu, kelembaban, derajat keasaman (pH), cahaya dan nutrisi yang ada pada media tanam jamur. Suhu ideal untuk

tumbuh kembang jamur berkisar diantara 25-30°C dan kelembaban udara berkisar diantara 60-90%. (Djarjah, 2001)

### 2.2.2 Protokol MQTT

Protokol *Message Queuing Telemetry Transport (MQTT)* merupakan protokol pengiriman data yang menggunakan metode *publish-subscribe*. Protokol dirancang agar mudah untuk diimplementasikan. Protokol MQTT dapat digunakan dalam komunikasi *machine-to-machine (M2M)* dan konteks *Internet of Things (IoT)*. Pola pengiriman pesan *publish-subscribe* membutuhkan *broker* untuk mengatur lalu lintas pengiriman data. *Broker* pada protokol MQTT bertanggung jawab untuk mendistribusikan pesan ke *client* yang meminta data yang dibutuhkan. Pengiriman pesan pada protokol MQTT menggunakan topik. Setiap *subscriber* meminta data kepada *publisher* maka *broker* akan mengatur pengiriman data yang sesuai dengan topik yang diminta oleh *subscriber*. Sistem monitoring didasarkan pada teknologi *IoT* dan terdiri dari tiga bagian penting: *MQTT Client Publisher*, *Server* atau *Broker* dan *MQTT Client Subscriber*. (Satria, 2014)



**Gambar 2.1 Mekanisme pengiriman data protokol MQTT**

Pada Gambar 2.1 merupakan mekanisme pengiriman data menggunakan protokol MQTT, pada sistem terdiri dari sensor sebagai *publisher*, server berperan sebagai *broker* untuk mengatur pengiriman data, serta terdapat *mobile device* dan laptop berperan sebagai *subscriber* untuk menerima data. Dalam sistem, sensor akan mempublish data kemudian akan diterima oleh *broker* terlebih dahulu dan pengiriman data akan diteruskan ke *subscriber* sesuai dengan topik yang diminta. Pada protokol MQTT terdapat beberapa level *QoS (Quality of Service)* yang digunakan untuk menjamin pengiriman data berjalan dengan baik. *QoS level 0* menjamin pengiriman pesan hanya sekali. *QoS level 1* akan memastikan pengiriman pesan paling tidak dilakukan satu kali. *QoS level 2* akan memastikan

pesan diterima satu kali. Pada level ini akan dipastikan bahwa pesan pasti diterima dan menjamin tidak akan terjadi duplikasi pada pesan yang terkirim.

### 2.2.3 Website

Website merupakan sekumpulan halaman web yang menampilkan informasi data yang didalamnya terdapat file dokumen yang ditulis menggunakan format *HTML (HyperText Markup Language)*. Halaman web berisi informasi data teks, audio maupun video ditransfer dari *server web* ke *client* atau *web browser* menggunakan protokol *Hypertext Transfer Protocol (HTTP)* (Jasmadi, 2015). *Website* bersifat statis dan dinamis, *website* statis mempunyai informasi tetap dan jarang berubah. Sedangkan pada *website* dinamis memiliki informasi yang dapat berubah-ubah dan informasi yang terdapat dalam *website* bersifat interaktif. (Amsyah, 2005)

#### 2.2.3.1 Web Server

*Web server* adalah software yang terdapat pada *server* yang berfungsi untuk menerima permintaan data dari klien seperti *web browser* berupa halaman web dengan menggunakan protokol HTTP maupun dengan menggunakan protokol HTTPS. Kemudian *web server* akan mengirimkan kembali permintaan data kepada klien dalam bentuk halaman web menggunakan format *HTML*. (Marisa, 2017). Pada umumnya dalam suatu halaman web yang dikirimkan oleh *web server* terdiri dari berbagai macam jenis data meliputi data gambar, video, audio dan teks. Terdapat berbagai macam jenis *web server* yang sering digunakan diantaranya *Apache Web Server*, *Apache Tomcat*, *Lighttpd*, *Zeus Web Server*, *Sun Java System Web Server* dan *Microsoft Windows Server*. (Ahmad, 2017)

### 2.2.4 THINGSBOARD

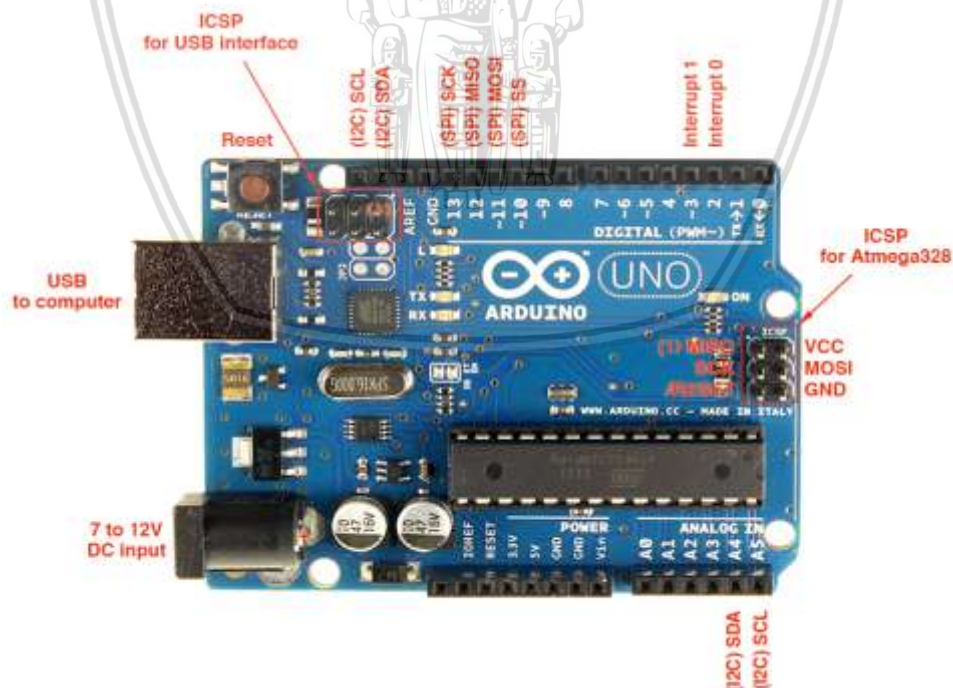
*Thingsboard* merupakan *platform Internet of Things (IoT)* yang bersifat open source. *Thingsboard* dikembangkan oleh *thingsboard.io* dan dapat digunakan sebagai platform manajemen device, pengumpulan data, visualisasi data berbasis website dan pengembangan berbagai macam *project* yang berbasis *Internet of Things (IoT)*. *Thingsboard* dapat diimplementasikan di berbagai macam sistem operasi seperti Windows, Linux (Ubuntu dan Centos), Mac OS serta dapat diimplementasikan secara langsung pada mikrokontroler seperti Raspberry Pi. Pada sisi pengiriman data, *thingsboard* mendukung pengiriman data real-time dan dapat menggunakan beberapa macam protokol komunikasi *IoT* seperti *MQTT*, *COAP*, dan *HTTP*. Keunggulan dari *platform thingsboard* adalah dapat digunakan untuk manajemen banyak *device* yang terhubung pada satu *server* dan memonitoring masing-masing *device* untuk menampilkan data dari masing-masing *device* yang terhubung via *thingsboard web user interface*. Contoh penerapan



thingsboard dalam IoT adalah sistem *Smart Farming, Fleet Tracking, Smart Metering dan Smart Energy*. (Thingsboard.io, 2017)

### 2.2.5 ARDUINO UNO

Arduino UNO merupakan mikrokontroller yang berbasis pada ATmega328. Pada mikrokontroller Arduino terdapat 14 pin digital input dan output. Pada 6 pin yang ada dapat digunakan untuk output PWM, 6 input analog, sebuah osilator Kristal 16 MHz, sebuah port USB, sebuah power jack DC 12 volt, sebuah ICSP header, dan sebuah tombol reset yang ditunjukkan pada Gambar 2.2. Bahasa pemrograman yang digunakan pada mikrokontroller Arduino yaitu C/C++. Dalam penelitian ini Arduino digunakan sebagai pengirim data dari sensor. Mikrokontroller Arduino digunakan dalam sistem ini sebagai ADC (*Analog to Digital Converter*) untuk merubah nilai pengeluaran dari sensor yang berupa analog menjadi *digital*. Pemilihan Arduino Uno pada pengimplementasian sistem dikarenakan Arduino merupakan sistem mikrokontroller yang relatif mudah dan cepat, Arduino menggunakan bahasa pemrograman yang sangat simpel dan daya yang sangat minim jika dibandingkan dengan ADC atau mikrokontroler lain, sehingga Arduino lebih mudah di pelajari dan di implementasikan pada sistem apapun. (Arduino.cc, 2015)



**Gambar 2.2 Arduino Uno**

Sumber : (Arduino.cc, 2015)

### 2.2.6 DHT – 22

Pada sistem ini sensor DHT-22 digunakan untuk membaca data suhu dan kelembaban yang dihubungkan dengan mikrokontroller Arduino. Sensor DHT-22 memiliki 3 pin yaitu pin VCC, Data dan GND yang ditunjukkan pada Gambar 2.3. Sensor DHT-22 membutuhkan sumber daya sebesar 5V. Data dari hasil pengukuran sensor DHT-22 sudah berupa sinyal digital dan sensor dapat mendeteksi suhu dan kelembaban dalam area sekitar 20 meter. Sensor DHT-22 mempunyai akurasi lebih baik dengan tingkat error sebesar  $\pm 2\%$  dibandingkan dengan sensor suhu dan kelembaban lainnya seperti sensor DHT-11 dan LM35. (Utama, 2016). Sensor DHT-22 memiliki rentang pembacaan data suhu adalah  $-40^{\circ}\text{C} - 80^{\circ}\text{C}$  dan rentang pembacaan data kelembaban pada sensor adalah  $0 - 100\% \text{ RH}$ .



**Gambar 2.3 Sensor DHT-22**

Sumber : (tokoteknologi.co.id, 2018)

### 2.2.7 Modul ESP8266-01

Pada sistem ini modul ESP8266-01 digunakan sebagai modul pengiriman data menggunakan jaringan nirkabel terhubung pada mikrokontroller Arduino Uno melalui koneksi pin yang ditunjukkan pada Gambar 2.4. Modul ini membutuhkan catu daya sebesar 3.3V dan memiliki 3 mode yaitu : Mode *Station* yang dapat digunakan sebagai *client* saja, Mode *Access Point*, dan Mode *Both* yang dapat digunakan sebagai *client* dan *access point* sekaligus. Pada modul ini terdapat prosesor, memori penyimpanan data dan GPIO dimana jumlah pin yang ada tergantung dengan jenis ESP8266 yang akan digunakan yang ditunjukkan pada tabel 2-1. (Widyaman, 2016). Pada modul ini menggunakan *AT-Command* sebagai *firmware defaultnya*, selain itu terdapat beberapa *Firmware SDK* yang

dapat digunakan oleh modul ini yang berbasis *opensource* yang diantaranya adalah *NodeMCU* dengan menggunakan bahasa pemrograman *LUA*, *MicroPython* dengan menggunakan bahasa pemrograman *python* dan *AT Command* dengan menggunakan perintah *AT command*.



**Gambar 2.4. ESP8266-01**

Sumber : (<https://www.sparkfun.com/>, 2016)

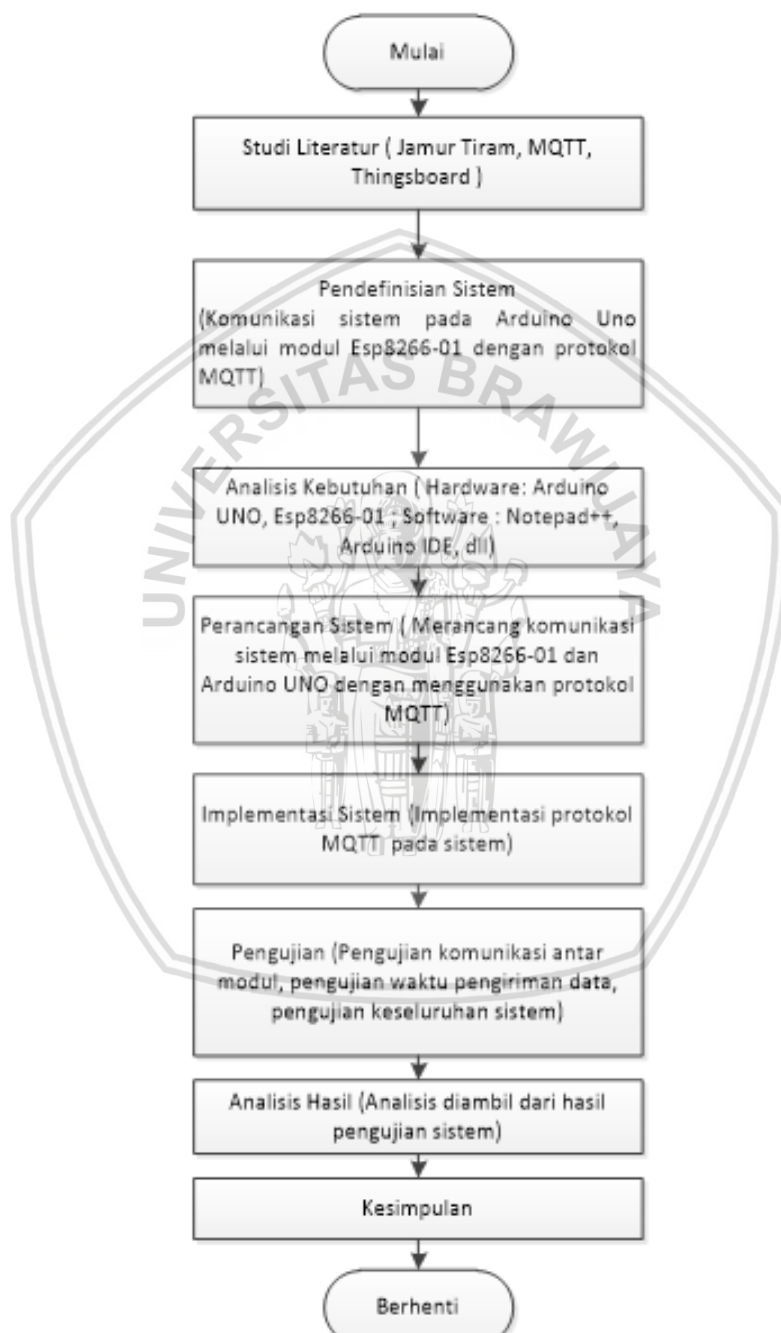
**Tabel 2-2 Pin Modul ESP8266-01**

No	Pin Modul ESP8266-01
1	VCC (3.3V)
2	RX
3	RST (RESET)
4	GPIO 0
5	CH_PD
6	GPIO 2
7	TX
8	GND (GROUND)

Pada Tabel 2-1 merupakan pin yang ada pada modul ESP8266-01, pin yang digunakan untuk terhubung dengan mikrokontroller Arduino adalah pin VCC digunakan sebagai sumber tegangan dari modul, pin RX dan pin TX digunakan untuk transfer data, dan pin GND digunakan sebagai ground dari modul.

### BAB 3 METODOLOGI PENELITIAN

Pada Bab ini akan menjelaskan alur pelaksanaan penelitian dari sistem yang akan dibuat. Tahapan penelitian yang akan dilakukan dapat digambarkan dalam bentuk *flowchart* seperti pada gambar 3.1 :



**Gambar 3.1 Alur Pelaksanaan Penelitian**

### 3.1 Studi Literatur

Pada penulisan penelitian yang dilakukan, studi literatur digunakan sebagai teori penguat serta landasan dasar dalam penelitian. Literatur yang digunakan pada penelitian ini meliputi:

1. Budidaya Jamur Tiram
2. *Message Queuing Telemetry Transport* (MQTT)
3. Website
4. Thingsboard
5. Mikrokontroller Arduino Uno
6. Modul ESP8266-01

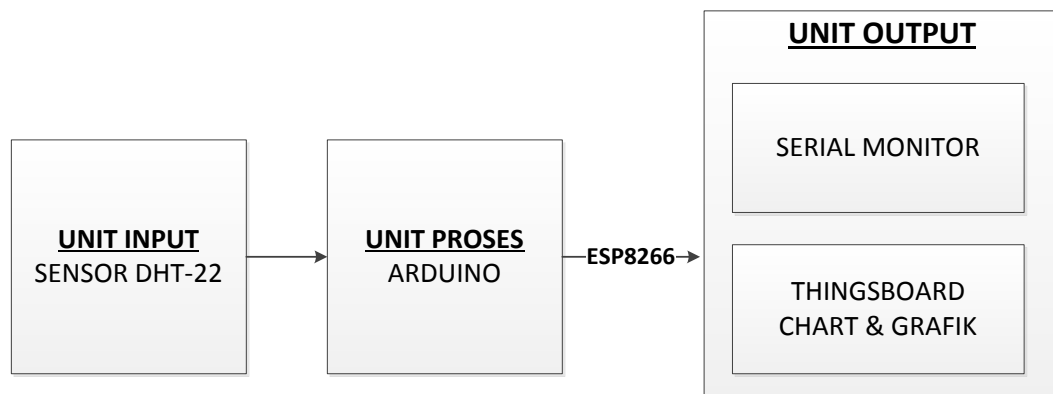
### 3.2 Analisis Kebutuhan

Kebutuhan yang diperlukan dalam proses pembangunan sistem meliputi :

1. Kebutuhan Fungsional
  - a. Sistem dapat membaca data dari sensor DHT-22 berupa data suhu dan kelembaban
  - b. Sistem mampu menampilkan nilai suhu dan kelembaban dari sensor dalam bentuk *chart* dan grafik
  - c. Hasil pengiriman data dapat dilihat melalui web server thingsboard.io
  - d. Jaringan yang digunakan oleh sistem merupakan jaringan *wireless*
2. Kebutuhan Non Fungsional
  - a. Pengiriman data pada sistem menggunakan modul *Wi-fi* ESP8266-01
  - b. Sensor yang digunakan untuk mendeteksi suhu dan kelembaban adalah sensor DHT-22
  - c. Jarak maksimal penerimaan sinyal dari *access point* ke modul *Wi-fi* ESP8266-01 adalah 100 meter
  - d. Sistem dapat bekerja secara langsung dengan menggunakan sumber listrik tanpa terhubung dengan komputer.

### 3.3 Perancangan Sistem

Pada bagian perancangan sistem adalah perancangan skematik dari perangkat keras serta perangkat lunak sistem. Pada unit input terdapat satu buah sensor, yaitu sensor DHT-22. Pada bagian unit proses menggunakan mikrokontroller Arduino Uno yang digunakan untuk mengolah data masukan dari sensor DHT-22 dan selanjutnya mengirimkan data menggunakan modul *Wi-fi* ESP8266-01. Pada unit output untuk menampilkan data suhu dan kelembaban berupa grafik melalui *web server thingsboard.io*



**Gambar 3.2 Diagram Blok Sistem**

### 3.4 Implementasi Sistem

Pada bagian implementasi sistem adalah melakukan penerapan sistem dengan menggunakan mikrokontroler Arduino Uno untuk mengolah data dari input sensor DHT-22 berupa nilai data suhu dan kelembaban. Pengiriman data pada sistem menggunakan modul *Wi-fi* ESP8266-01. Selanjutnya data tersebut akan dikirimkan ke *web server* thingsboard dengan menggunakan protokol MQTT. Setelah data diterima oleh server, data akan ditampilkan dalam bentuk *chart* dan grafik.

### 3.5 Pengujian

Pengujian pada sistem ini dilakukan dengan cara mengukur jumlah waktu yang dibutuhkan pada saat sensor mendeteksi data suhu dan kelembaban dan data diproses menggunakan Arduino Uno hingga data dapat tampil di *web server Thingsboard*. Kehandalan kinerja sistem diukur untuk mengetahui kemampuan sistem dalam pengiriman ke *server* dengan menggunakan protokol MQTT.

Tujuan dari proses pengujian yang dilakukan pada sistem ini adalah agar sistem dapat menunjukkan bahwa sistem dapat bekerja dengan baik. Pengujian yang dilakukan antara lain meliputi :

1. Pengujian kinerja sensor
2. Pengujian *delay* waktu pengiriman data menggunakan protokol MQTT
3. Pengujian fungsionalitas sistem

Setelah proses pengujian sistem dan mendapatkan hasil dari serangkaian pengujian yang telah dilakukan, maka hasil dari pengujian tersebut akan dianalisis untuk mengetahui bagaimana kinerja dari sistem yang dibuat dan dapat ditarik sebuah kesimpulan dari penelitian yang telah dibuat.



## BAB 4 REKAYASA KEBUTUHAN

### 4.1 Deskripsi Umum

Bab ini menjelaskan tentang rekayasa kebutuhan yang diperlukan untuk perancangan dan implementasi sistem meliputi kebutuhan perangkat keras dan perangkat lunak sistem, batasan sistem dan kegunaan sistem. Sehingga dalam perancangan dan implementasi sistem monitoring pada rumah jamur menggunakan jaringan nirkabel berbasis protokol MQTT dapat berfungsi dengan baik.

#### 4.1.1 Perspektif Umum

Pada sistem ini dinilai dapat bekerja jika sistem mampu mengambil data nilai suhu dan kelembaban menggunakan sensor DHT-22 dan data diproses oleh Arduino Uno dan selanjutnya data dikirim menggunakan modul ESP8266-01 melalui jaringan nirkabel ke web server Thingsboard.io menggunakan protokol MQTT. Data nilai suhu dan kelembaban pada web server Thingsboard ditampilkan dalam bentuk grafik.

#### 4.1.2 Kegunaan

Dengan adanya sistem monitoring suhu dan kelembaban pada rumah jamur menggunakan protokol MQTT, sistem monitoring ini berguna agar petani jamur dapat mengetahui perubahan suhu dan kelembaban yang terjadi dalam rumah jamur dan dapat menentukan tindakan yang diperlukan selanjutnya untuk mendapatkan hasil budidaya jamur yang maksimal.

#### 4.1.3 Karakteristik Pengguna

Karakteristik pengguna pada sistem ini adalah masyarakat yang membudidayakan jamur atau disebut dengan petani jamur dimana petani jamur dapat memantau perubahan suhu dan kelembaban yang terjadi pada rumah jamur.

#### 4.1.4 Lingkungan Operasi Sistem

Pada dokumentasi ini kebutuhan lingkungan yang mendukung bekerjanya sistem yaitu :

1. Sensor dan komponen perangkat keras Arduino Uno harus ditempatkan pada titik yang aman pada rumah jamur seperti terhindar dari tempat basah agar tidak merusak sistem.
2. Modul *Wi-fi* ESP8266-01 beroperasi pada frekuensi 2.4 Ghz jika terhalang tembok yang cukup tebal dapat mengganggu penerimaan sinyal dari *access point* sehingga komponen modul *Wi-fi* dan Arduino uno harus diletakkan pada tempat yang mempunyai sedikit celah terbuka pada rumah jamur.

#### 4.1.5 Batasan Sistem

Beberapa batasan yang terdapat pada sistem ini antara lain :

1. Qos (*Quality of Service*) yang digunakan pada protokol MQTT dalam sistem ini adalah Qos level 0 dan level 1 sesuai dengan platform thingsboard, yaitu dapat mengirimkan data minimal satu kali dan menerima *puback* dari MQTT broker.
2. Alamat dari *web server* sudah diketahui yaitu *demo.thingsboard.io*
3. Topik yang digunakan dalam MQTT berupa data suhu dan kelembaban
4. *Broker* yang digunakan pada sistem adalah *broker mosquito*
5. Pengiriman data dalam sistem menggunakan modul *Wi-fi* ESP8266-01
6. Web server menggunakan *platform thingsboard*

#### 4.1.6 Asumsi dan Ketergantungan

Beberapa asumsi dan ketergantungan pada sistem ini antara lain :

1. Pengiriman data suhu dan kelembaban pada sistem dapat berjalan dengan baik jika penempatan komponen Arduino Uno dan modul *Wi-fi* ESP8266-01 tidak terhalang oleh benda yang dapat mengganggu penerimaan sinyal dari *access point*.
2. Data dari sensor dapat dikirimkan jika perangkat *publisher* terhubung dengan jaringan lokal komputer.
3. Sensor dan modul ESP8266-01 dapat berfungsi dengan baik jika *library* pada program yang digunakan sudah sesuai dengan modul tersebut.

#### 4.2 Rekayasa Kebutuhan

Rekayasa kebutuhan pada sistem menjabarkan semua kebutuhan dengan tujuan agar sistem berjalan dengan baik, meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak, kebutuhan komunikasi dan kebutuhan lainnya pada sistem.

##### 4.2.1 Kebutuhan Antarmuka Pengguna

Kebutuhan antar muka pengguna pada sistem ini berupa grafik pada *web server* Thingsboard yang datanya dikirimkan oleh *publisher* dengan menggunakan protokol MQTT.

##### 4.2.2 Kebutuhan Komunikasi

Kebutuhan komunikasi pada sistem ini menggunakan jaringan nirkabel dengan menggunakan modul *Wi-fi* ESP8266-01 dan disambungkan dengan mikrokontroller Arduino Uno. Untuk kebutuhan komunikasinya digunakan library Wi-FiEsp dan library PubSubClient untuk protokol MQTT. Setelah proses inisialisasi

selesai, maka device akan terhubung dengan server yang sudah diketahui alamat IP nya. Protokol pengiriman data yang digunakan pada sistem ini adalah protokol MQTT.

#### 4.2.3 Kebutuhan Sistem

Untuk mendukung sistem dapat bekerja dengan baik dibutuhkan sensor yang dapat mendeteksi suhu dan kelembaban sebagai *input* data dari sistem. Pada sistem ini sensor yang digunakan adalah sensor DHT22. Untuk memenuhi kebutuhan komunikasi *wireless* pada sistem digunakan modul *Wi-Fi* ESP8266-01. Dan untuk mikrokontroler yang digunakan pada sistem ini adalah mikrokontroler Arduino Uno. Dasar penggunaan Arduino Uno dikarenakan mikrokontroler tersebut dapat mengakomodasi kebutuhan pin sensor DHT22 dan modul *Wi-Fi* ESP8266-01.

#### 4.2.4 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras pada sistem ini meliputi :



Gambar 4.1 Diagram Kebutuhan Perangkat Keras Sistem

1. Publisher :
  - a. Arduino Uno digunakan sebagai mikrontroller yang mengakomodasi pin dari sensor suhu dan kelembaban DHT22 dan modul *wireless* ESP8266-01 dan mengatur *input / output* dari rangkaian sistem.
  - b. Sensor DHT 22 digunakan sebagai membaca data pada sistem berupa nilai suhu dan kelembaban.
  - c. Modul ESP8266-01 digunakan untuk mengirimkan data pada sistem dengan menggunakan jaringan *wireless*.
2. Perangkat komputer digunakan untuk mengakses *web server thingsboard* yang berperan sebagai broker dan subscriber pada sistem dan menampilkan data melalui *web browser*.

#### 4.2.5 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak sistem meliputi :

1. Publisher :
  - a. Arduino IDE digunakan untuk merancang program dan menjalankan program pada mikrokontroler Arduino Uno
  - b. DHT library digunakan sebagai library dari sensor DHT22
  - c. Library PubSubClient digunakan sebagai library protokol MQTT
  - d. Library *Wi-FiEsp* digunakan sebagai library modul *Wi-Fi* ESP8266-01

2. Thingsboard

Thingsboard digunakan sebagai web server pada sistem untuk menampilkan data suhu dan kelembaban berupa grafik. Pada web server thingsboard terdapat MQTT Broker yang digunakan untuk mengatur proses pengiriman data oleh publisher dan subscriber. Broker yang digunakan adalah *Mosquitto broker v3.1*. Web server thingsboard juga berperan sebagai MQTT subscriber yang bertugas untuk meminta data dari publisher.

#### 4.2.6 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang harus terpenuhi sehingga sistem dapat bekerja dengan baik.

- a. Pembacaan data sensor DHT-22  
Sensor DHT-22 digunakan untuk membaca nilai data suhu dan kelembaban pada rumah jamur
- b. Pengiriman data menggunakan modul *Wi-fi* ESP8266-01  
Modul ESP8266-01 terhubung dengan access point dan mengirimkan data suhu dan kelembaban ke web server thingsboard
- c. Sistem dapat menampilkan dalam bentuk grafik  
Data suhu dan kelembaban dapat dimonitoring dalam bentuk grafik
- d. *Web server* yang digunakan adalah *web server* thingsboard.io
- e. Jaringan yang digunakan sistem adalah jaringan *wireless*

#### 4.2.7 Kebutuhan Non Fungsional

Kebutuhan non fungsional meliputi :

- a. Sensor DHT-22  
Sensor DHT-22 digunakan untuk membaca nilai suhu dan kelembaban sebagai inputan dari sistem, nilai yang dijadikan acuan dari sensor harus akurat dan perlu dilakukan kalibrasi sensor. Kalibrasi sensor dilakukan

dengan membandingkan hasil pembacaan sensor DHT-22 dengan alat dari pabrikan berupa termometer digital yang dapat membaca suhu dan kelembaban.

- b. Jarak maksimal penerimaan sinyal *Wi-Fi* dari *access point* ke modul *Wi-Fi* ESP8266-01 adalah 100 meter
- c. Sistem dapat bekerja secara langsung dengan menggunakan sumber listrik tanpa terhubung dengan komputer.
- d. Power supply pada sistem

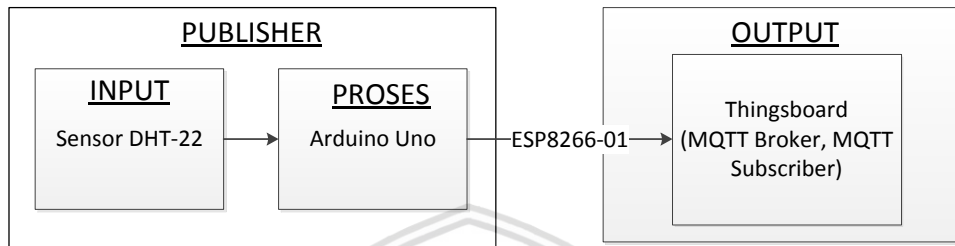
Sumber daya atau power supply pada sistem dapat menggunakan adaptor tanpa terhubung langsung pada komputer



## BAB 5 PERANCANGAN DAN IMPLEMENTASI

### 5.1 Perancangan Perangkat Keras

Pada sistem monitoring pada rumah jamur berbasis protokol MQTT terbagi menjadi 3 bagian yaitu *publisher*, *broker* dan *webserver*. Sistem secara keseluruhan ditunjukkan pada Gambar 5.1 berikut ini.



**Gambar 5.1 Perancangan umum sistem**

Pada gambar 5.1 bagian publisher terdiri dari sensor DHT-22 sebagai input data suhu dan kelembaban, selanjutnya data akan diproses melalui program pada mikrokontroller arduino dan data hasil proses akan dikirimkan menggunakan modul ESP8266-01 ke web server thingsboard menggunakan protokol MQTT. Pada proses pengiriman data pada sistem diatur oleh MQTT broker yang ada pada web server thingsboard.

#### 5.1.1 Perancangan Koneksi Sensor

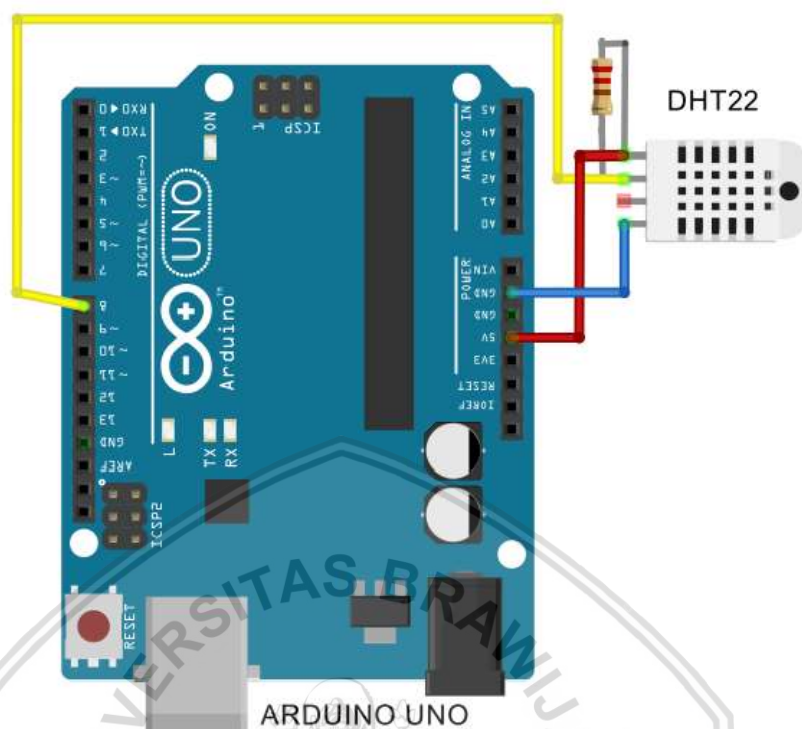
Pada sistem ini sensor yang akan digunakan adalah sensor DHT-22 sebagai input data suhu dan kelembaban. Sensor DHT-22 terhubung dengan mikrokontroller arduino uno melalui koneksi pin yang ditunjukkan oleh Gambar 5.2.



**Gambar 5.2 Sensor DHT-22**

Pada Gambar 5.2 merupakan gambar sensor DHT-22 yang memiliki beberapa pin. Pada pin 1 digunakan sebagai sumber tegangan (VCC) , pin 2 digunakan sebagai transfer data (DATA) dan pin 3 digunakan sebagai ground (GND).





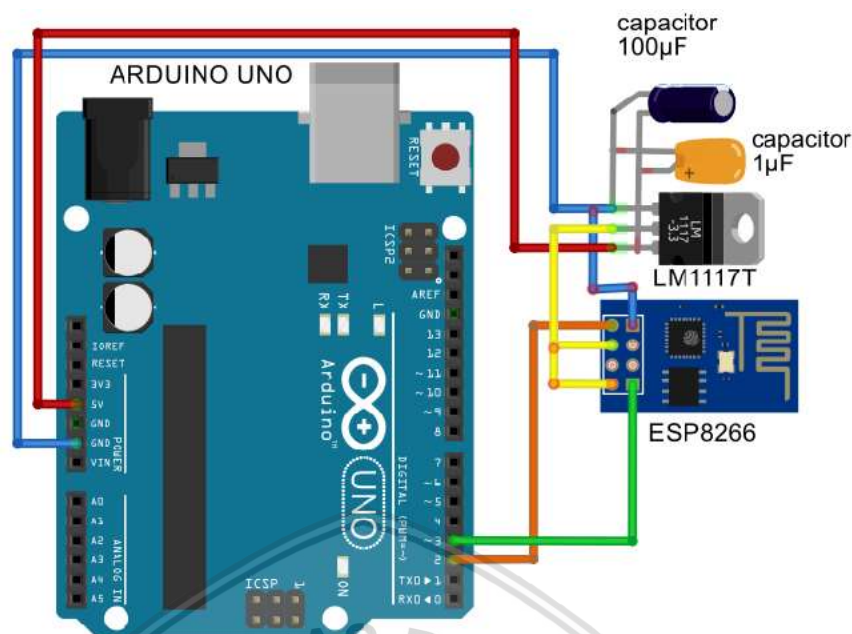
**Gambar 5.3 Koneksi sensor DHT-22 ke Arduino**

Pada gambar 5.3 menunjukkan koneksi pin out dari sensor DHT-22 ke Arduino adalah sebagai berikut :

1. pin 1 dengan kabel berwarna merah terhubung ke VCC 5 volt sebagai sumber tegangan,
2. pin 2 dengan kabel berwarna kuning terhubung ke pin 8 pada arduino sebagai transfer data suhu dan kelembaban, antara pin 1 dan pin 2 pada sensor DHT-22 ditambahkan resistor 10K untuk menghambat tegangan dari VCC 5 volt arduino dan selanjutnya pin 3 dengan kabel berwarna biru terhubung ke ground pada arduino.

### 5.1.2 Perancangan Koneksi Modul ESP8266-01

Modul ESP8266-01 digunakan sebagai modul pengiriman data pada sistem ini. Modul ESP8266-01 terhubung dengan *access point* yang terdapat pada lingkungan sekitar untuk mengirimkan data suhu dan kelembaban dari sensor DHT-22 yang sudah diproses oleh mikrokontroler arduino menuju ke webserver thingsboard untuk ditampilkan datanya secara keseluruhan seperti pada Gambar 5.4.



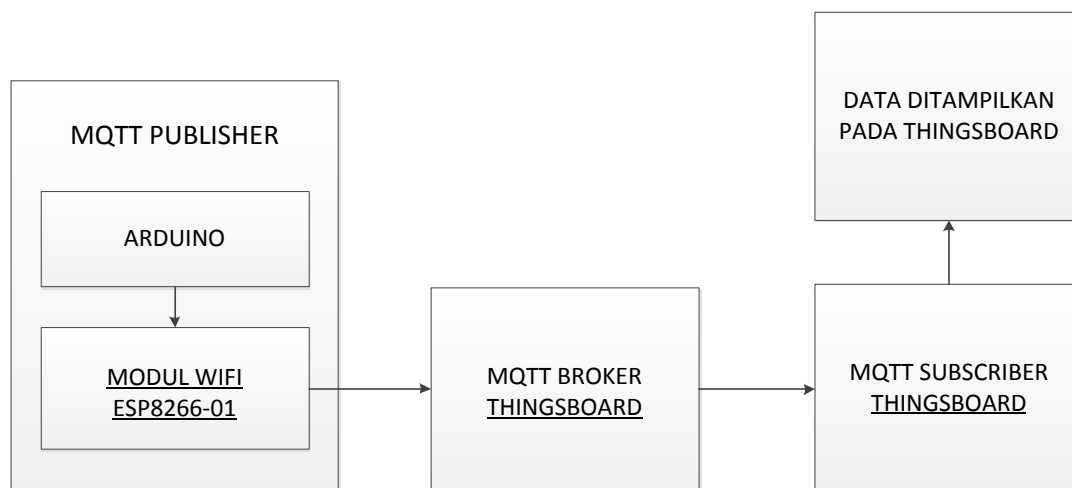
**Gambar 5.4 Modul ESP8266-01**

Pada gambar 5.4 modul ESP8266-01 terhubung dengan mikrokontroler arduino dengan rincian sebagai berikut.

1. pin TX terhubung pada Pin 3 Arduino, Pin RX terhubung dengan Pin 2 Arduino yang digunakan sebagai transceiver dan receiver untuk akses data antara modul ESP8266-01 dengan arduino.
2. Pin CH\_PD dan VCC terhubung pada regulator tegangan LM1117T pada pin output.
3. Pin GND (ground) terhubung dengan pin ground pada regulator LM1117T. Regulator LM1117T digunakan untuk menurunkan tegangan dari 5 volt ke 3.3 volt.
4. Pin input LM1117T dihubungkan dengan VCC 5 volt arduino dan pin ground LM1117T dihubungkan dengan ground pada arduino. Masing-masing antara VCC dan ground pada LM1117T ditambahkan Kapasitor 100 uF dan keramik kapasitor 1 uF.

### 5.1.3 Perancangan Pengiriman Data Menggunakan MQTT

Protokol MQTT pada sistem monitoring pada rumah jamur ini digunakan sebagai protokol pengiriman data suhu dan kelembaban dari MQTT publisher menuju webserver thingsboard yang berfungsi sebagai MQTT subscriber. Pada proses pengiriman data suhu dan kelembaban diatur oleh MQTT broker yang ada pada platform thingsboard. MQTT broker pada sistem ini menggunakan broker mosquitto 3.1 yang ditunjukkan pada Gambar 5.6.

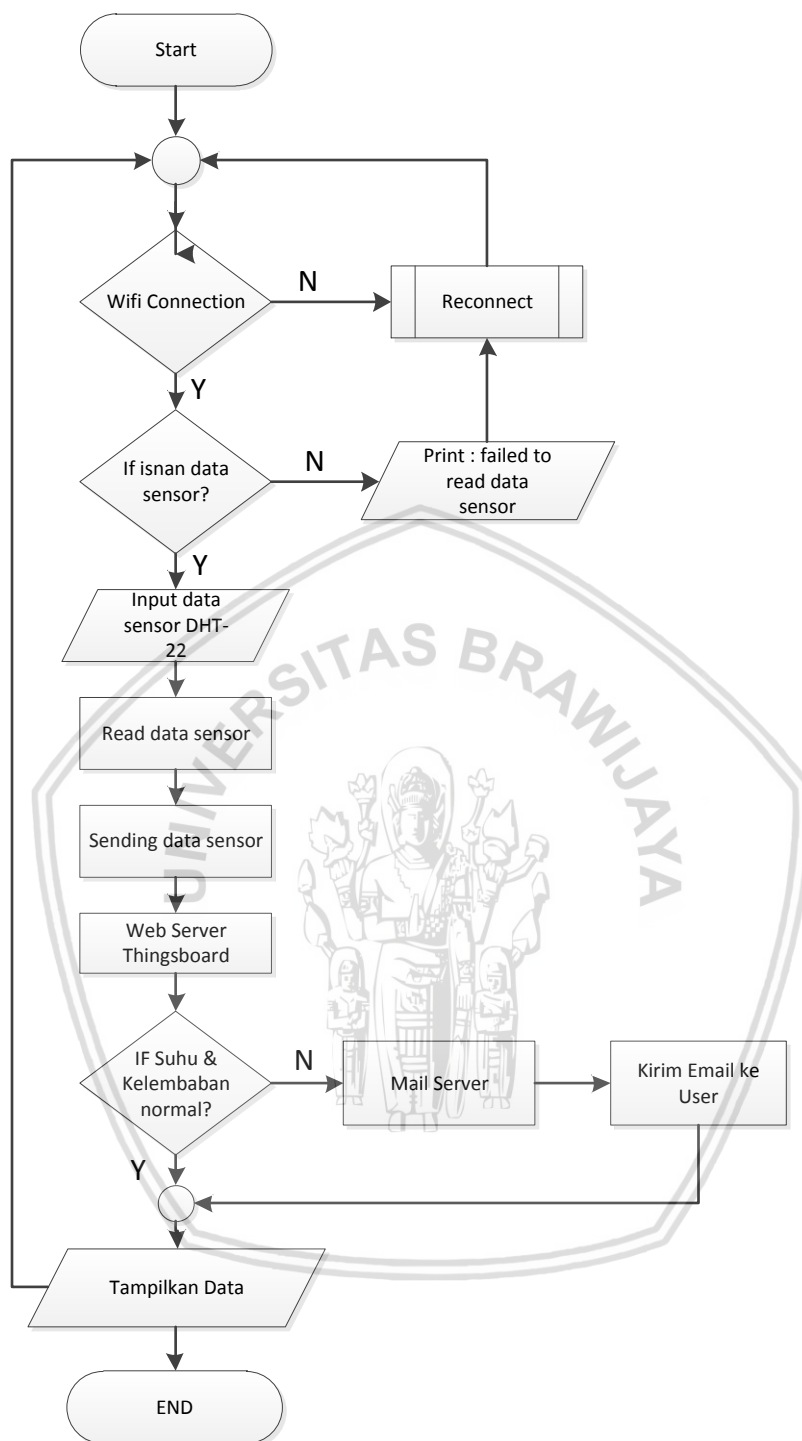


**Gambar 5.5 Perancangan MQTT**

Pada Gambar 5.5 merupakan perancangan MQTT pada sistem, *Thingsboard* pada sistem ini berfungsi sebagai *broker* yang mengatur distribusi data dan dapat juga berperan sebagai *subscriber* yang meminta topik data pada *publisher*. QoS (*Quality of Service*) pada protokol ini menggunakan QoS level 0 yaitu paling banyak satu data dapat dikirim dan QoS level 1 yaitu minimal satu data dapat dikirim. *Publisher* dapat mengirimkan data ke *server Thingsboard* dengan menggunakan akses token yang didapatkan dari *platform Thingsboard*. Pada bagian MQTT *publisher* pada sistem ini terdiri dari sensor DHT-22, Arduino, dan modul *Wi-fi* ESP8266 yang akan mengirimkan data menggunakan fungsi *payload*, fungsi tersebut dapat digunakan untuk mengirimkan data berbentuk *byte*. Semua data yang sudah diterima oleh server *Thingsboard* akan ditampilkan dalam bentuk *chart* dan grafik.

## 5.2 Perancangan Perangkat Lunak

Pada perancangan perangkat lunak digunakan untuk menggambarkan bagaimana proses dari sistem yang akan dibuat dan alur berjalannya sistem yang ditunjukkan pada Gambar 5.6.



**Gambar 5.6 Flowchart Sistem Monitoring Suhu dan Kelembaban**

Pada Gambar 5.6 merupakan diagram alir dari sistem monitoring pada rumah jamur menggunakan protokol MQTT. Sistem akan memulai program dan menghubungkan perangkat dengan jaringan *Wi-fi* terlebih dahulu. Jika sistem sudah terhubung dengan jaringan *Wi-fi*, selanjutnya sensor akan membaca data suhu dan kelembaban yang ada pada rumah jamur menggunakan sensor DHT-22, namun jika sensor gagal membaca data suhu dan kelembaban maka akan muncul

pesan pada serial monitor "*Print : Failed to read data sensor*" dan sistem akan memulai program dari awal. Data yang sudah dibaca oleh sensor kemudian dikirimkan ke web server thingsboard. Jika suhu dan kelembaban tidak sesuai dengan batas yang ditentukan pada rentang suhu 25-30°C dan kelembaban udara berkisar diantara 60-90%, sistem akan memberikan pesan peringatan ke pengguna melalui *mail server* ke alamat *email* pengguna.

### 5.2.1 Pembacaan Data Pada Sensor

Dalam proses pembacaan data sensor, harus dilakukan inisialisasi terlebih dahulu pada Arduino IDE dengan menggunakan *library* dari sensor. Sensor terhubung dengan mikrokontroller Arduino melalui koneksi pin. Sensor mendeteksi suhu dan kelembaban pada rumah jamur dan data tersebut akan diproses oleh mikrokontroller Arduino dan data dikirim dengan menggunakan modul ESP8266-01 yang terhubung dengan akses poin ke webserver thingsboard menggunakan protokol MQTT.

### 5.2.2 Pengiriman Data

Dalam pengiriman data pada sistem ini menggunakan modul ESP8266-01. Pin TX terhubung pada pin 2 Arduino dan Pin RX terhubung pada pin 3 Arduino. Kedua pin tersebut digunakan untuk proses transfer data antara kedua perangkat tersebut. Pada sistem ini digunakan fungsi *payload* untuk mengirimkan paket data ke web server. Fungsi *payload* tersebut berfungsi untuk menyimpan hasil pembacaan data dari sensor agar dapat dikirim dalam bentuk *byte* ke *server Thingsboard*.

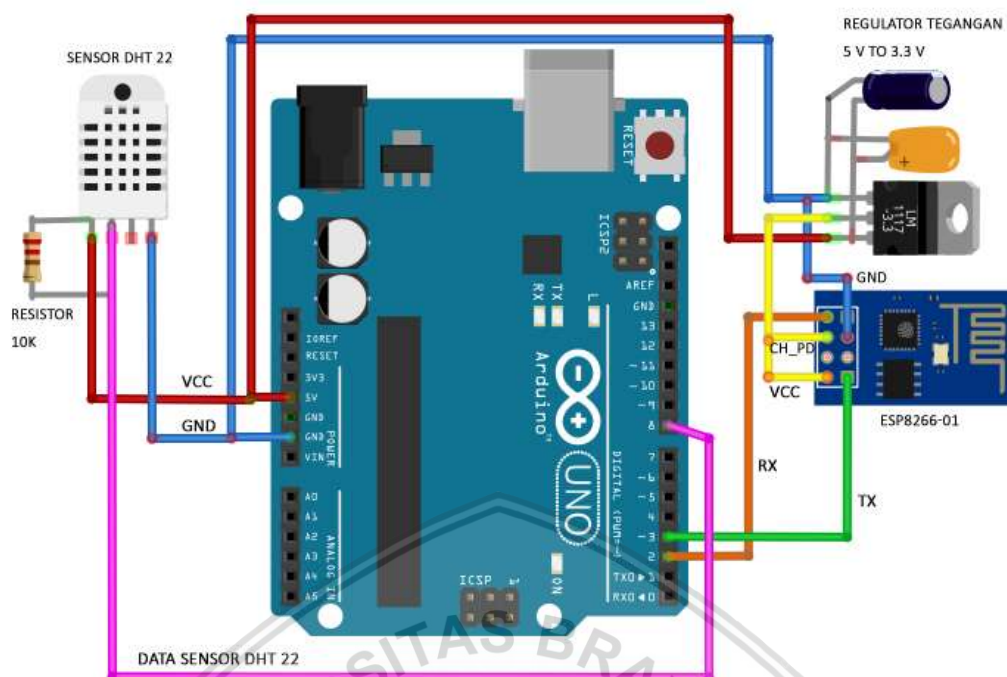
Pada bagian pengiriman data pada sistem ini menggunakan topik "*temperature*" dan "*humidity*" agar dapat berjalan sesuai dengan format yang ada pada *server thingsboard*. Topik tersebut disimpan dalam fungsi *payload* dan akan dikirimkan menggunakan format *json* menuju ke *server Thingsboard*.

## 5.3 Implementasi Sistem

### 5.3.1 Implementasi Hardware

Pada sistem monitoring pada rumah jamur ini menggunakan beberapa *hardware* diantaranya mikrokontroller Arduino sebagai pemroses data, modul ESP8266-01 sebagai pengirim data dan sensor DHT-22 sebagai pembaca data suhu dan kelembaban yang ada pada rumah jamur.





**Gambar 5.7 Skema Rangkain Sensor Suhu dan Kelembaban**

Pada Gambar 5.7 sensor DHT-22 pin 1 dengan kabel berwarna merah terhubung ke VCC 5 volt sebagai sumber tegangan, pin 2 dengan kabel berwarna merah muda terhubung ke pin 8 pada arduino sebagai transfer data suhu dan kelembaban, antara pin 1 dan pin 2 pada sensor DHT-22 ditambahkan resistor 10K untuk menghambat tegangan dari VCC 5 volt arduino dan selanjutnya pin 3 dengan kabel berwarna biru terhubung ke ground pada arduino.

Pada modul ESP8266-01 terhubung dengan mikrokontroler arduino dengan pin TX terhubung pada Pin 3 Arduino, Pin RX terhubung dengan Pin 2 Arduino yang digunakan sebagai transceiver dan receiver untuk akses data antara modul ESP8266-01 dengan arduino. Pin CH\_PD dan VCC terhubung pada regulator tegangan LM117T pada pin output. Pin GND (ground) terhubung dengan pin ground pada regulator LM117T. Regulator LM117T digunakan untuk menurunkan tegangan dari 5 volt ke 3.3 volt. Pin input LM117T dihubungkan dengan VCC 5 volt arduino dan pin ground LM117T dihubungkan dengan ground pada arduino. Masing-masing antara VCC dan ground pada LM117T ditambahkan Kapasitor 100 uF dan keramik kapasitor 1 uF.





**Gambar 5.8 Rangkaian Sensor Suhu dan Kelembaban**

Pada bagian proses dari sistem, digunakan fungsi *time* untuk mengatur waktu eksekusi proses. Pada fungsi tersebut digunakan *interval* proses pengiriman data adalah 1 detik. Sehingga data dari sistem dapat dikirimkan ke *server* setiap 1 detik. Pada sistem ini juga digunakan fungsi *micros* untuk menghitung waktu yang dibutuhkan sistem mulai dari sensor membaca data hingga dikirimkan melalui modul ESP8266 ke *server thingsboard*.

### 5.3.2 Implementasi Software

Pada proses implementasi software diperlukan langkah awal untuk melakukan inisialisasi hardware yang digunakan seperti sensor DHT-22 dan modul ESP8266-01 ke Arduino IDE menggunakan library yang bertujuan agar fungsi dari hardware yang digunakan dapat berjalan pada program.

**Tabel 5-1 Kode Sumber Library Arduino**

Algoritme 1 : Library IDE Arduino	
1	#include "DHT.h"
2	#include <Wi-FiEspClient.h>
3	#include <Wi-FiEsp.h>
4	#include <Wi-FiEspUdp.h>
5	#include <PubSubClient.h>
6	#include "SoftwareSerial.h"

Pada Tabel 5-1 ditampilkan library yang digunakan pada Arduino IDE.

1. Library DHT.h merupakan library yang digunakan sebagai fungsi dari sensor DHT-22 agar dapat menjalankan fungsi pembacaan suhu dan kelembaban.

2. Library *Wi-FiEspClient.h* , *Wi-FiEsp.h* , *Wi-FiEspUdp.h* merupakan library dari modul ESP8266-01 agar dapat menjalankan fungsinya mengirimkan data suhu dan kelembaban ke webserver.
3. Library *SoftwareSerial* digunakan untuk mengatur komunikasi serial pada Arduino
4. Library untuk protokol MQTT adalah library *PubSubClient*. Library tersebut merupakan library untuk menjalankan fungsi dari protokol MQTT.

Setelah proses inialisasi library selesai, selanjutnya adalah proses untuk modul ESP8266-01 terhubung ke *access point* menggunakan SSID dan password yang sudah ditentukan. Kemudian inialisasi program untuk mengakses ke webserver thingsboard menggunakan fungsi *Token* sesuai dengan *device* yang ada pada *webserver* thingsboard seperti pada Tabel 5-2. Satu *ID Token* digunakan untuk satu *device* sehingga pada saat proses pengiriman data tidak terdapat resiko salah pengiriman data pada *device* yang lain.

**Tabel 5-2 Kode Sumber Inialisasi Access Point dan Token Thingsboard**

Algoritme 2 : Inialisasi Access Point dan Token Thingsboard	
1	#define WIFI_AP "innovate"
2	#define WIFI_PASSWORD "akuanaksoleh"
3	#define TOKEN "62nI9Mx503mrU4O6QCQ7"

### 5.3.2.1 Pembacaan Data Sensor

Pada proses pembacaan data sensor dilakukan inialisasi pin terlebih dahulu. Sensor DHT-22 terhubung pada pin 6 mikrokontroller Arduino. Pada Tabel 5-3 merupakan kode program inialisasi pin sensor DHT-22 dan mendefinisikan sensor yang digunakan yaitu sensor DHT-22.

**Tabel 5-3 Kode Sumber Definisi Pin dan Tipe Sensor DHT-22**

Algoritme 3 : Definisi Pin dan Tipe Sensor DHT-22	
1	#define DHTPIN 6
2	#define DHTTYPE DHT22

Selanjutnya proses untuk terhubung ke web server thingsboard "demo.thingsboard.io" digunakan kode program seperti pada Tabel 5-4.

**Tabel 5-4 Kode Sumber Definisi alamat server Thingsboard**

Algoritme 4 : Definisi alamat server Thingsboard	
1	char server[] = "demo.thingsboard.io";

Kemudian dilanjutkan proses inialisasi *Wi-Fi* ESP agar modul ESP8266-01 dapat digunakan dan menggunakan komunikasi serial pada pin 2 untuk Receiver dan pin 3 untuk Transceiver. Fungsi "*dht*" digunakan untuk mengakses data dari

sensor dan fungsi PubSubClient digunakan untuk menjalankan protokol MQTT seperti pada Tabel 5-5.

**Tabel 5-5 Kode Sumber Inisialisasi fungsi pada program**

Algoritme 5 : Inisialisasi fungsi pada program	
1	<code>WiFiEspClient espClient;</code>
2	<code>DHT dht(DHTPIN, DHTTYPE);</code>
3	<code>PubSubClient client(espClient);</code>
4	<code>SoftwareSerial soft(2, 3); // RX, TX</code>

### 5.3.2.2 Proses Modul ESP8266-01 terhubung ke jaringan

Untuk dapat berkomunikasi antara Modul ESP8266-01 dengan mikrokontroller Arduino menggunakan standar baudrate 9600. Jika proses komunikasi serial gagal maka akan muncul pesan pada serial monitor bahwa "Modul Wi-fi tdk terdeteksi" ditunjukkan pada Tabel 5-6.

**Tabel 5-6 Kode Sumber Inisialisasi Modul ESP8266-01 ke Arduino**

Algoritme 6 : Inisialisasi Modul ESP8266-01 ke Arduino	
1	<code>void InitWiFi()</code>
2	<code>{</code>
3	<code>soft.begin(9600);</code>
4	<code>WiFi.init(&amp;soft);</code>
5	<code>if (WiFi.status() == WL_NO_SHIELD) {</code>
6	<code>Serial.println("Modul Wi-Fi tdk terdeteksi");</code>
7	<code>while (true);</code>
8	<code>}</code>

Selanjutnya jika proses komunikasi serial antara modul ESP8266-01 dan mikrokontroller Arduino berhasil maka modul ESP8266-01 akan mencoba untuk terhubung ke *access point* menggunakan fungsi *WIFI\_AP* untuk mengakses SSID dan *WIFI\_PASSWORD* untuk mengakses password dari SSID seperti pada Tabel 5-7.

**Tabel 5-7 Kode Sumber Koneksi Modul ESP8266-01 ke jaringan Wi-fi**

Algoritme 7 : Koneksi Modul ESP8266-01 ke jaringan Wi-fi	
1	<code>Serial.println("Menghubungkan ke Access Point ...");</code>
2	<code>// attempt to connect to Wi-Fi network</code>
3	<code>while ( status != WL_CONNECTED) {</code>
4	<code>Serial.print("Menghubungkan ke SSID: ");</code>
5	<code>Serial.println(WIFI_AP);</code>

6	// Connect to WPA/WPA2 network
7	status = <i>WiFi</i> .begin( <i>WIFI_AP</i> , <i>WIFI_PASSWORD</i> );
8	delay(500);
9	}
10	Serial.println("Terhubung ke Access Point");
11	}

Pada Tabel 5-7 merupakan kode program untuk menghubungkan modul ESP8266-01 ke jaringan *WiFi*, nama perangkat SSID dan passwordnya akan dicocokkan melalui variabel *WIFI\_AP* dan *WIFI\_PASSWORD*. Jika sudah benar maka modul ESP8266-01 akan terhubung ke jaringan *wi-fi* dan pada serial monitor akan muncul pesan "Terhubung ke Access Point".

**Tabel 5-8 Kode Sumber Penghitungan waktu proses pada program**

Algoritme 8 : Penghitungan waktu proses pada program	
1	void loop() {
2	Mulai = micros();
3	if ( micros() - lastSend > 1000000
4	lastSend = micros();
5	Selesai = micros();
6	Serial.print("Waktu : ");
7	Serial.print(Selesai-Mulai);
8	Serial.println(" micros");

Pada Tabel 5-8 merupakan program untuk melihat waktu yang dibutuhkan dari proses pembacaan data sensor hingga data selesai dikirim. Pada program, variabel "*Mulai*" berfungsi sebagai waktu awal ketika Arduino akan menjalankan perintah pada program. Variabel tersebut menggunakan tipe data *unsigned long* yang dapat menyimpan data dengan jumlah yang besar. Penggunaan fungsi *micros* dalam sistem akan menghitung waktu mulai dari proses pertama program berjalan hingga selesai yang diinisialisasikan dengan variabel "*Selesai*". Akhir proses dari sistem ini adalah ketika data yang diolah oleh Arduino sudah dikirimkan menggunakan modul *Wi-fi* ESP8266. Perintah *Selesai = micros()*. Digunakan untuk menghentikan penghitungan waktu pada sistem. Interval pengiriman data pada sistem adalah 1 detik jika dikonversikan dalam satuan milisekon menjadi 1.000.0000 milisekon. Cara penghitungan waktu pada program dilakukan dengan mencari selisih antara waktu selesai dengan waktu mulai (*Selesai-Mulai*). Hasil dari proses penghitungan waktu tersebut akan ditampilkan pada serial monitor saat program berjalan.

### 5.3.2.3 Pengiriman Data Sensor

Pada bagian pengiriman data, nilai dari sensor DHT-22 didefinisikan pada sistem variabel *h* untuk kelembaban dan variabel *t* untuk suhu.

**Tabel 5-9 Kode Sumber Pengambilan nilai data sensor**

Algoritme 9 : Pengambilan Nilai Data Sensor	
1	<code>float h = dht.readHumidity();</code>
2	<code>float t = dht.readTemperature();</code>
3	<code>if (isnan(h)    isnan(t)) {</code>
4	<code>    Serial.println("Gagal membaca data sensor!");</code>
5	<code>    return;</code>
6	<code>}</code>
7	<code>Serial.print("Kelembaban: ");</code>
8	<code>Serial.print(h);</code>
9	<code>Serial.print(" %\t");</code>
10	<code>Serial.print("Suhu: ");</code>
11	<code>Serial.print(t);</code>
12	<code>Serial.print(" *C ");</code>

Pada Tabel 5-9 variabel *t* dan *h* menggunakan tipe data *float* yang mendukung penggunaan angka desimal. Variabel *h* merupakan nilai data kelembaban sedangkan variabel *t* merupakan data nilai suhu pada sensor DHT-22. Pada program ditambahkan fungsi *isnan*. Melalui fungsi *isnan*, variabel *h* dan *t* akan dicek apakah pada saat data diambil nilainya kosong atau *false*. Apabila nilai data kosong maka pada serial monitor akan muncul pesan yang bertuliskan "*Gagal membaca data sensor!*". Selanjutnya jika proses pengambilan data berhasil maka nilai data akan ditampilkan pada serial monitor.

**Tabel 5-10 Kode Sumber Payload pengiriman paket data**

Algoritme 10 : Payload pengiriman paket data	
1	<code>String payload = "{";</code>
2	<code>    payload += "\"temperature\":"; payload += temperature;</code>
3	<code>    payload += ",";</code>
4	<code>    payload += "\"humidity\":"; payload += humidity;</code>
5	<code>    payload += "}";</code>

Pada Tabel 5-10 merupakan proses pengiriman data menggunakan protokol MQTT yang format *json*. Untuk pengiriman data dalam format *json*, maka akan digunakan fungsi *payload*. Fungsi tersebut digunakan untuk menyimpan data dalam bentuk *byte*.



**Tabel 5-11 Kode Sumber Mengirim payload ke Thingsboard**

Algoritme 11 : Mengirim Payload ke Thingsboard	
1	char attributes[100];
2	payload.toCharArray( attributes, 100 );
3	client.publish( "v1/devices/me/telemetry", attributes );
4	Serial.println( attributes );
5	}

Pada Tabel 5-11 untuk mempublish topik menggunakan fungsi *payload* menggunakan perintah *client.publish* pada program, topik yang dapat digunakan pada server thingsboard adalah "*v1/devices/me/telemetry*", *attributes* merupakan topik *default* dari thingsboard. Fungsi *payload* pada program digunakan untuk menyimpan data dari sensor yang dirubah dalam bentuk paket terdahulu sebelum dikirimkan dalam topik *telemetry*. Setelah data suhu dan kelembaban diterima oleh thingsboard menggunakan topik *telemetry*, data akan ditampilkan dalam bentuk *chart* dan grafik.

**5.3.3 Implementasi Sistem Pemberitahuan Suhu dan Kelembaban**

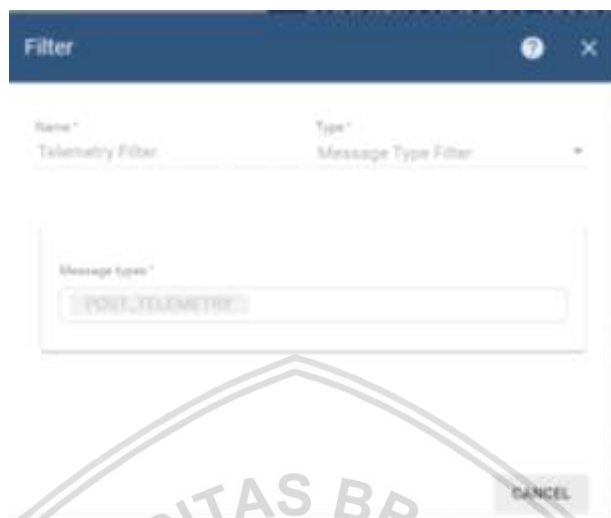
Pada proses pemberitahuan suhu dan kelembaban pada sistem ini akan mendeteksi suhu dan kelembaban yang ada pada rumah jamur. Suhu standar yang ditentukan berkisar diantara 25-30°C dan kelembaban udara berkisar diantara 60%-90%. Jika suhu kurang atau melebihi standar sistem akan mengirimkan email kepada pengguna sistem dengan menggunakan fungsi Rules yang ada pada web server Thingsboard. Sebelum membuat fungsi rules pada thingsboard harus ditentukan ZoneId dan ZoneType dengan yang ada pada Server Attribute di device yang dipakai oleh penulis yaitu device ARDUINO UNO.



**Gambar 5.9 Server Attribute Thingsboard**

Pada Gambar 5.9 merupakan Server Attribute yang ada pada Thingsboard. ZoneId dan ZoneType digunakan untuk menentukan zona dan tipe yang akan dipakai pada

fungsi Rules. Dalam fungsi rules terdapat beberapa filter yang akan dibuat diantaranya Message Type Filter yang digunakan untuk mengirim data suhu dan kelembaban ke mail server seperti pada Gambar 5.10.



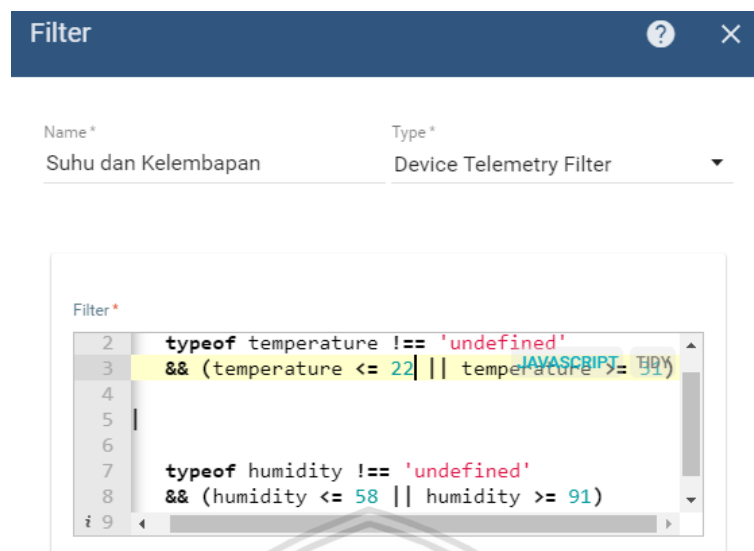
**Gambar 5.10 Message Telemetry Filter**

Selanjutnya menentukan Attribute filter yang digunakan untuk menentukan ZoneType pada sistem seperti pada Gambar 5.11.



**Gambar 5.11 Menentukan ZoneType Filter pada Thingsboard**

Fungsi selanjutnya adalah menentukan batas atas dan batas bawah suhu dan kelembaban yang sesuai dengan standar yaitu berkisar diantara 25-30°C dan kelembaban udara berkisar diantara 60%-90% pada fungsi Telemetry Filter seperti pada Gambar 5.12.



**Gambar 5.12 Menentukan Suhu dan Kelembapan pada Sistem**

Selanjutnya adalah membuat fungsi Alarm Prosesor untuk mencatat waktu jika terjadi perubahan suhu dan kelembapan dan menentukan ZoneId pada sistem seperti pada Gambar 5.13.



**Gambar 5.13 Alarm Prosesor pada Sistem**

Setelah ZoneId ditentukan dan waktu perubahan suhu sudah diketahui, selanjutnya jika terjadi perubahan suhu dan kelembapan yang tidak sesuai dengan standar yang telah ditentukan maka sistem akan memberikan pemberitahuan kepada pengguna melalui email dengan menyertakan waktu dan data suhu dan kelembapan yang sudah dicatat oleh sistem. Pada sistem ini penulis menggunakan plugin mail server yang kompatibel dengan server Thingsboard yaitu plugin mail server SendGrid. Dengan mendaftarkan alamat email terlebih dahulu kemudian melakukan konfigurasi plugin pada Thingsboard dengan menyertakan alamat email yang akan digunakan seperti pada Gambar 5.14.

Plugin action	
Alarm Suhu dan Kelembapan	Send Mail Action

Send flag (empty or isNewAlarm, isExistingAlarm, isClearedAlarm, isNewOrClearedAlarm)  
isNewAlarm

From template\*

nugraha.pangestu@gmail.com

To template\*

nugraha.pangestu@gmail.com

CC template

BCC template

Subject template\*

\$alarmId

Body template\*

\$alarmBody

**Gambar 5.14 Konfigurasi Email Plugin**

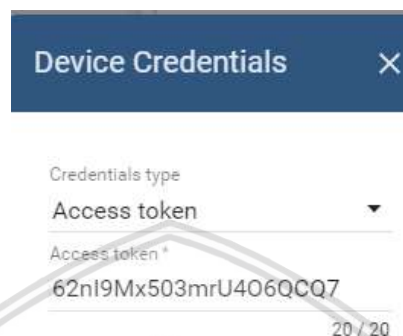
### 5.3.4 Hasil Pengiriman Data ke Web Server

Pada proses untuk menampilkan data ke *server Thingsboard*, terlebih dahulu mengakses server pada alamat <http://demo.thingsboard.io>. Kemudian masuk pada menu *Device* dan menentukan *device* yang akan digunakan. Pada sistem ini *device* yang digunakan adalah *device "Aduino UNO"* pada seperti pada Gambar 5.15.



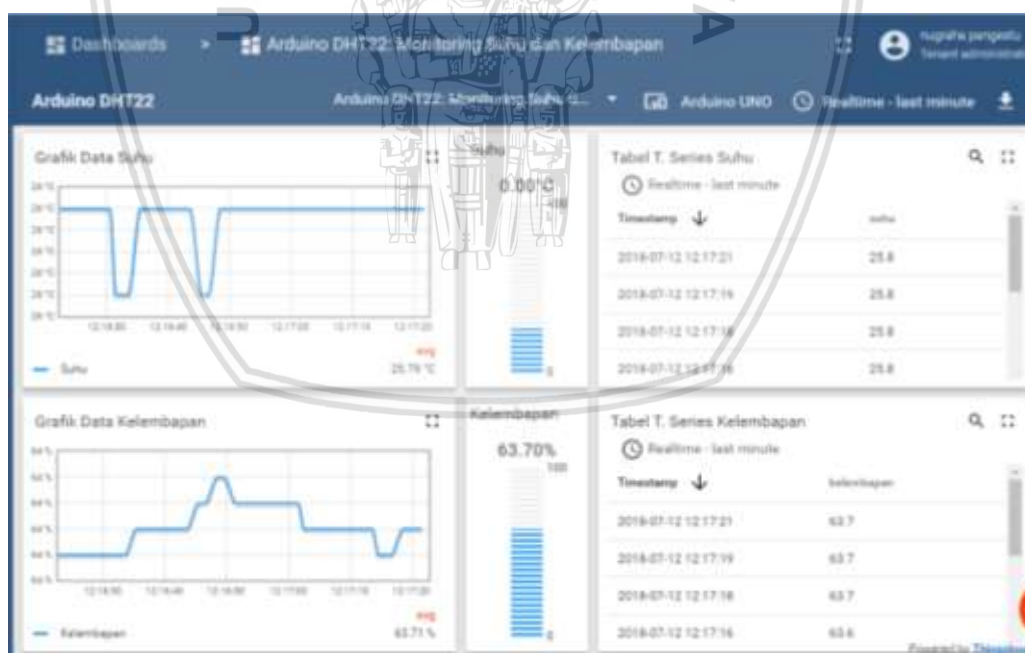
**Gambar 5.15 Tampilan Devices pada Server Thingsboard**

Pada program untuk dapat terhubung dengan *device* yang ada pada *server Thingsboard*, diperlukan *id\_token*. *ID\_token* didapatkan dari *device credentials* yang ada pada Thingsboard ditunjukkan pada Gambar 5.16. Setelah mendapatkan *id\_token* dari server, pada bagian program Arduino perlu dicantumkan *id\_token* tersebut sehingga data yang akan dikirimkan dapat masuk ke *device* yang sudah ditentukan.



**Gambar 5.16 Akses Token Devices**

Selanjutnya setelah dapat mengakses device pada thingsboard, untuk menampilkan data suhu dan kelembaban dapat menggunakan widget berupa grafik dan chart yang ada pada menu *Dashboards* seperti pada Gambar 5.17.



**Gambar 5.17 Tampilan Data Menggunakan Grafik dan Chart**

Pada Gambar 5.17 merupakan tampilan dari menu *Dashboard Thingsboard* yang menunjukkan nilai data suhu dan kelembaban berupa grafik dan chart sehingga jika terjadi perubahan nilai data suhu dan kelembaban dapat diketahui dengan mudah. Selanjutnya jika terjadi perubahan suhu yang tidak sesuai dengan

standar maka sistem memberikan pemberitahuan melalui email ke pengguna seperti pada Gambar 5.18.



**Gambar 5.18 Email Pemberitahuan Suhu dan Kelembaban**





## BAB 6 PENGUJIAN DAN ANALISIS

### 6.1 Pengujian

#### 6.1.1 Pengujian Pembacaan Data Sensor

##### 6.1.1.1 Tujuan Pengujian

Tujuan dari pengujian yang akan dilakukan adalah untuk mengukur dan mengetahui bagaimana pembacaan sensor sebagai input data dari sistem monitoring pada rumah jamur menggunakan protokol MQTT.

##### 6.1.1.2 Prosedur Pengujian

Pada bagian pengujian ini, sistem akan diuji untuk mengetahui tingkat keberhasilan dalam membaca data dari sensor. Proses pengujian dilakukan dengan cara sebagai berikut:

1. Sistem dijalankan hingga terhubung dengan server Thingsboard,
2. Pada sensor DHT-22 didekatkan pada nyala api dari korek untuk mengetahui perubahan suhu dan kelembaban yang terjadi.
3. Nilai pada sensor DHT-22 dibandingkan dengan termometer buatan pabrik untuk mengetahui akurasi pembacaan data sensor.
4. Melihat perubahan suhu dan kelembaban pada serial monitor.

##### 6.1.1.3 Hasil Pengujian

Pengujian ini menghasilkan perubahan nilai suhu dan kelembaban seperti pada Gambar 6.1 dan 6.2.

```
Data suhu dan kelembapan
Kelembapan: 81.50 %    Suhu: 23.70 *C Mengirim data suhu & kelembapan : [23.70,81.50]
Waktu : 980852 micros
Data suhu dan kelembapan
Kelembapan: 81.50 %    Suhu: 23.70 *C Mengirim data suhu & kelembapan : [23.70,81.50]
Waktu : 669760 micros
Data suhu dan kelembapan
Kelembapan: 81.60 %    Suhu: 23.70 *C Mengirim data suhu & kelembapan : [23.70,81.60]
Waktu : 915588 micros
Data suhu dan kelembapan
Kelembapan: 81.60 %    Suhu: 23.70 *C Mengirim data suhu & kelembapan : [23.70,81.60]
Waktu : 669760 micros
```

**Gambar 6.1 Data Sensor sebelum didekatkan pada nyala api**

```

Data suhu dan kelembapan
Kelembapan: 80.90 %    Suhu: 24.90 *C Mengirim data suhu & kelembapan : [24.90,80.90]
Waktu : 693344 micros
Data suhu dan kelembapan
Kelembapan: 80.90 %    Suhu: 24.90 *C Mengirim data suhu & kelembapan : [24.90,80.90]
Waktu : 959640 micros
Data suhu dan kelembapan
Kelembapan: 80.90 %    Suhu: 24.90 *C Mengirim data suhu & kelembapan : [24.90,80.90]
Waktu : 667808 micros
Data suhu dan kelembapan
Kelembapan: 80.70 %    Suhu: 24.90 *C Mengirim data suhu & kelembapan : [24.90,80.70]
Waktu : 914396 micros
Data suhu dan kelembapan

```

**Gambar 6.2 Data Sensor sesudah didekatkan pada nyala api**

Pada Gambar 6.1 dan 6.2 dapat diketahui bahwa sensor dapat mendeteksi perubahan suhu dan kelembaban pada saat sensor ketika didekatkan pada nyala api hal ini menunjukkan sensor dapat mendeteksi suhu dan kelembaban dengan baik. Selanjutnya adalah membandingkan nilai data dari sensor DHT-22 dengan termometer buatan pabrik seperti pada gambar 6.3 dan 6.4.



**Gambar 6.3 Suhu dan kelembaban pada termometer**

```

Data suhu dan kelembapan
Kelembapan: 84.70 %    Suhu: 28.40 *C Mengirim data suhu & kelembapan : [28.40,84.70]
Waktu : 503312 micros
Data suhu dan kelembapan
Kelembapan: 84.70 %    Suhu: 28.40 *C Mengirim data suhu & kelembapan : [28.40,84.70]
Waktu : 473176 micros
Data suhu dan kelembapan
Kelembapan: 84.60 %    Suhu: 28.50 *C Mengirim data suhu & kelembapan : [28.50,84.60]
Waktu : 914136 micros

```

**Gambar 6.4 Suhu dan kelembaban dari sensor DHT-22**

Pada Gambar 6.3 dan 6.4 merupakan hasil perbandingan data suhu dan kelembaban dari termometer buatan pabrik dan sensor DHT-22, pada termometer menunjukkan kelembaban 83% dan suhu 29.3 °C, sedangkan pada sensor DHT-22 menunjukkan kelembaban 84.6 % dan suhu 28.5°C. Dari perbandingan pengukuran suhu dan kelembaban pada termometer dan sensor DHT-22 didapatkan selisih pengukuran kelembaban 1.6 % dan suhu 0.8°C. Dari selisih pengukuran suhu dan kelembaban yang didapatkan antara termometer dengan sensor DHT-22 menunjukkan bahwa akurasi dari pengukuran sensor sangat baik.

### 6.1.2 Pengujian Delay pada MQTT

#### 6.1.2.1 Tujuan Pengujian

Pengujian ini dilakukan untuk mengetahui jumlah waktu pengiriman data dari sistem monitoring pada rumah jamur menggunakan protokol MQTT.

#### 6.1.2.2 Prosedur Pengujian

Pengujian delay ini dilakukan dengan cara mengurangi waktu data yang tampil di *server Thingsboard* dengan waktu ketika data selesai diproses pada serial monitor Arduino. Pada pengujian ini ada beberapa hal yang perlu diabaikan yaitu koneksi jaringan internet yang tidak stabil karena modul *Wi-Fi ESP8266-01* sangat sensitif terhadap naik turunnya sinyal dari jaringan *Wi-Fi* sehingga dapat mempengaruhi waktu pengiriman data. Prosedur pengujian delay waktu pengiriman data adalah sebagai berikut:

1. Menjalankan sistem hingga dapat mendeteksi suhu dan kelembaban pada Arduino dan melihat apakah perangkat sudah terhubung pada jaringan *Wi-Fi*,
2. Mencatat waktu saat sistem mulai mendeteksi data dari sensor,
3. Melihat jumlah waktu dari proses sistem pada serial monitor,
4. Delay MQTT diperoleh dari waktu data yang ditampilkan pada Thingsboard, dikurangi dengan data waktu proses sistem dengan satuan mikrosekon.

#### 6.1.2.3 Hasil Pengujian Delay pada MQTT

Pada pengujian delay waktu pengiriman data pada thingsboard, pengujian dilakukan sebanyak 20 kali dengan waktu yang berbeda. Dari percobaan pengiriman data sebanyak 20 kali, didapatkan rata-rata waktu yang dibutuhkan sensor mendeteksi adanya perubahan suhu dan kelembaban hingga data ditampilkan di Thingsboard adalah 794181.6 mikrosekon atau 0,7941816 detik, jika dilakukan pembulatan maka waktu yang dibutuhkan adalah 0,79 detik. Hasil pengujian delay pengiriman data dapat dilihat pada Tabel 6-1.

**Tabel 6-1 Hasil Pengujian Waktu Ekekusi Program pada Arduino**

No.	Waktu Pengujian	Waktu Eksekusi (dalam mikrosekon)
1	19:56:00	673064
2	19:58:00	686444
3	20:10:41	917684
4	20:17:00	773988

5	20:19:20	997460
6	20:33:00	672780
7	07:29:16	917412
8	07:33:13	787340
9	09:36:00	917704
10	10:44:19	687784
11	10:49:38	675016
12	11:55:28	684700
13	11:59:50	767640
14	14:04:40	998320
15	14:08:10	674864
16	15:10:04	770876
17	15:12:20	671264
18	16:21:15	917860
19	16:30:01	771844
20	16:45:49	919588
RATA-RATA		794181.6

Kemudian dilakukan pengujian selanjutnya dengan berdasarkan pada waktu data dari sensor DHT-22 ditampilkan ke web server Thingsboard. Terdapat delay pada saat data dikirim hingga ditampilkan pada Thingsboard. Pengujian delay tersebut dapat dilihat pada Tabel 6-2.

**Tabel 6-2 Hasil Pengujian Waktu Data yang Tampil di Thingsboard**

No.	Waktu Pengujian	Waktu yg Tampil di Thingsboard
1	19:56:00	19:56:02
2	19:58:00	19:58:02
3	20:10:41	20:10:42
4	20:17:00	20:17:02
5	20:19:20	20:19:22
6	20:33:00	20:33:02
7	07:29:16	07:29:18
8	07:33:13	07:33:15

9	09:36:00	09:36:02
10	10:44:19	10:44:21
11	10:49:38	10:49:40
12	11:55:28	11:55:30
13	11:59:50	11:59:52
14	14:04:40	14:04:42
15	14:08:10	14:08:12
16	15:10:04	15:10:06
17	15:12:20	15:12:22
18	16:21:15	16:21:17
19	16:30:01	16:30:03
20	16:45:49	16:45:51
RATA-RATA		1,95 detik

Dari Tabel 6-2 dapat diketahui bahwa pada sistem terdapat *delay* ketika sensor mulai membaca data hingga data sudah ditampilkan pada *server Thingsboard*. Dari 20 kali pengujian yang telah dilakukan menghasilkan rata-rata delay 1,95 detik. Dari Tabel 6-1 dan Tabel 6-2 dapat diketahui bahwa pada proses pengiriman data terdapat selisih waktu. Selisih tersebut merupakan waktu pengiriman data dari *publisher* menuju ke *server Thingsboard*. Selisih tersebut dapat diketahui dengan cara mengurangi waktu yang tampil pada *server Thingsboard* dengan waktu pemrosesan data yang tampil pada serial monitor pada Arduino. Untuk mengetahuinya dapat dilihat di Tabel 6-3.

**Tabel 6-3 Penghitungan delay pengiriman data dari Thingsboard dikurangi waktu pada serial monitor**

No	Waktu Pengujian	Waktu Eksekusi (dalam mikrosekon)	Waktu yg tampil di Thingsboard	Selisih dalam detik
1	19:56:00	673064	19:56:02	1,326
2	19:58:00	686444	19:58:02	1,313
3	20:10:41	917684	20:10:42	1,082
4	20:17:00	773988	20:17:02	1,226
5	20:19:20	997460	20:19:22	1,002
6	20:33:00	672780	20:33:02	1,327

7	07:29:16	917412	07:29:18	1,082
8	07:33:13	787340	07:33:15	1,212
9	09:36:00	917704	09:36:02	1,082
10	10:44:19	687784	10:44:21	1,312
11	10:49:38	675016	10:49:40	1,324
12	11:55:28	684700	11:55:30	1,315
13	11:59:50	767640	11:59:52	1,232
14	14:04:40	998320	14:04:42	1,001
15	14:08:10	674864	14:08:12	1,325
16	15:10:04	770876	15:10:06	1,229
17	15:12:20	671264	15:12:22	1,328
18	16:21:15	917860	16:21:17	1,082
19	16:30:01	771844	16:30:03	1,228
20	16:45:49	919588	16:45:51	1,080
RATA-RATA		794181.6 Micros	1,205 sec	

Dari Tabel 6-3 dapat diketahui bahwa setelah dilakukan 20 kali pengujian menghasilkan rata-rata delay pengiriman data dalam sistem monitoring pada rumah jamur menggunakan protokol MQTT adalah 1,205 detik. Jika dilakukan pembulatan angka maka didapatkan hasil rata-rata menjadi 1,20 detik. Pada sistem ini waktu yang dibutuhkan untuk mendeteksi perubahan suhu dan kelembaban pada sensor hingga data ditampilkan pada server Thingsboard adalah 1,20 detik.

### 6.1.3 Pengujian Fungsionalitas Sistem

#### 6.1.3.1 Tujuan Pengujian

Tujuan dari pengujian ini adalah untuk mendapatkan hasil yang optimal dari perancangan sistem monitoring pada rumah jamur menggunakan protokol MQTT dengan pengujian secara keseluruhan pada sistem.

#### 6.1.3.2 Prosedur Pengujian

Pengujian hasil pada sistem dilakukan dengan cara sebagai berikut:

1. Sistem diaktifkan hingga tersambung ke jaringan *Wi-fi*,
2. Sensor DHT 22 akan mendeteksi suhu dan kelembaban didalam ruangan rumah jamur,



3. Pada Thingsboard akan ditampilkan perubahan data dalam bentuk *chart* dan grafik.

#### 6.1.3.3 Hasil Pengujian

Pada pengujian keseluruhan sistem, perangkat keras dari sistem ditempatkan didalam ruang rumah jamur untuk mendeteksi suhu dan kelembapan dan datanya dikirimkan melalui jaringan *Wi-Fi* menggunakan protokol MQTT ke server Thingsboard, terlihat bahwa ada perubahan suhu dan kelembapan yang tercatat dalam *chart* dan grafik seperti pada Gambar 6.5.



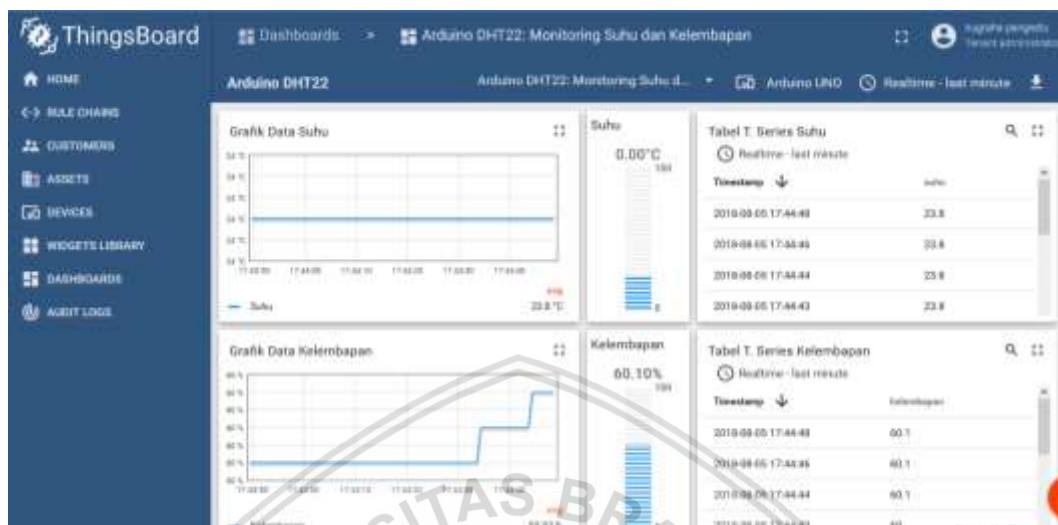
**Gambar 6.5 Kondisi rumah jamur**

Pada Gambar 6.5 merupakan kondisi didalam rumah jamur. Pada saat melakukan pengujian, sistem mendeteksi suhu dan kelembapan yang ada pada rumah jamur dan mengirimkan data ke server Thingsboard.



**Gambar 6.6 Hardware sistem monitoring suhu dan kelembaban**

Pada Gambar 6.6 merupakan *hardware* dari sistem yang digunakan untuk mendeteksi suhu dan kelembaban yang ada pada rumah jamur dan digunakan juga untuk mengirimkan datanya menggunakan jaringan *Wi-fi*.



**Gambar 6.7 Tampilan data suhu dan kelembaban pada Thingsboard**

Pada Gambar 6.7 merupakan tampilan layar ketika Thingsboard dapat menampilkan data yang dideteksi oleh sensor dengan baik. Perubahan data suhu dan kelembaban yang terjadi dapat dilihat melalui tampilan grafik dan *chart* pada Thingsboard.

## 6.2 Analisis

Dari hasil pengujian yang dilakukan sebanyak 20 kali pengiriman data pada sistem monitoring pada rumah jamur menggunakan protokol MQTT, didapatkan rata-rata waktu pengiriman data adalah 1,20 detik yang dihitung sejak sensor membaca data hingga ditampilkan pada server Thingsboard dalam bentuk *chart* dan grafik. Dari hasil pengujian tersebut dapat diketahui bahwa dalam proses pengiriman data suhu dan kelembaban pada sistem ke server Thingsboard dapat berubah-ubah karena terdapat beberapa hal yang dapat mempengaruhi waktu pengiriman data seperti kendala jaringan *internet*, kondisi server Thingsboard dan *bandwidth* yang digunakan.

## BAB 7 KESIMPULAN DAN SARAN

### 7.1 Kesimpulan

Pada perancangan sistem monitoring pada rumah jamur menggunakan protokol MQTT yang dibuat penulis diperoleh kesimpulan berdasarkan rumusan masalah sebagai berikut:

1. Perancangan pada sistem ini menggunakan satu sensor yaitu sensor DHT-22 yang dihubungkan dengan mikrokontroler Arduino dan modul *Wi-fi* ESP8266-01 yang terhubung dengan jaringan *Wi-fi* dan digunakan untuk mengirimkan data ke *server Thingsboard*.
2. Berdasarkan beberapa kali pengujian sistem yang telah dilakukan, rata-rata waktu yang dibutuhkan sistem untuk mengirim data suhu dan kelembaban ke *server Thingsboard* adalah 1,20 detik, sehingga pengimplementasian protokol MQTT dalam sistem monitoring ini cukup efisien.
3. Hasil yang didapatkan dari perancangan sistem yaitu sistem dapat membaca perubahan suhu dan kelembaban pada rumah jamur menggunakan protokol MQTT ke *server Thingsboard* dan data dapat ditampilkan dalam bentuk *chart* dan grafik.

### 7.2 Saran

Untuk menghasilkan sistem yang lebih baik lagi diperlukan beberapa pengembangan pada sistem yang telah dibuat oleh penulis sebagai berikut :

1. Pada sistem ditambahkan otomasi untuk menstabilkan suhu dan kelembaban ketika terjadi perubahan kondisi pada rumah jamur,
2. Penggunaan *Thingsboard* dapat dimanfaatkan untuk memonitoring beberapa area sekaligus sehingga dapat digunakan sebagai sistem monitoring dengan lebih optimal,

## DAFTAR PUSTAKA

- Ahmad, A. (2017). Retrieved from <https://bukainfo.com/pengertian-web-server-sebenarnya/>
- Amsyah, Z. (2005). *Manajemen Sistem Informasi*. Jakarta: Gramedia Pustaka Utama.
- Andi Stanford-Clark, H. L. (2013). MQTT For Sensor Networks (MQTT-SN).
- Arduino.cc. (2015). *Arduino & genuino products*. Retrieved 12 30, 2015, from Arduino: <https://www.arduino.cc/en/Main/ArduinoBoardUno>
- Aziz, D. E. (2017). Retrieved from <http://www.kangerik.id/pengertian-fungsi-serta-cara-kerja-web-server/>
- Budioko, T. (2016). Sistem Monitoring Suhu Jarak Jauh Berbasis Internet of Things Menggunakan Protokol MQTT.
- Cahyana, dkk. (1999). *Jamur Tiram*. Jakarta: Swadaya.
- Chazali, Syammahfuz . (2010). *Usaha Jamur Tiram Skala Rumah Tangga*. Jakarta: Swadaya.
- Djarajah, Nunung Marlina. (2001). *Budidaya Jamur Tiram*. Yogyakarta: Kanisus.
- Faisal, E. K. (2015, 6). *Socket*. Retrieved 12 12, 2015, from Microsoft: <https://msdn.microsoft.com/id-id/library/mt154252.aspx>
- <https://www.sparkfun.com/>. (2016). Retrieved from <https://www.sparkfun.com/>: <https://www.sparkfun.com/products/13678>
- Jasmadi. (2015). In *Langsung Praktek, Cari Uang di Internet dari Blog dan AdSense*. Elex Media Komputindo.
- Joko Nugroho. (2014). SISTEM MONITORING PENDETEKSI SUHU DAN KELEMBAPAN PADA RUMAH JAMUR BERBASIS MIKROKONTROLLER AT-MEGA 328.
- Kautsar, M. S. (2014). RANCANG BANGUN APLIKASI PEMANTAU DATA WIRELESS SENSOR NETWORK UNTUK PERINGATAN DINI TERHADAP BANJIR.
- Kuctso. (2015, 12 8). *hal ini menjadi sulit untuk startup kecil atau perusahaan skala kecil untuk menemukan programmer*. Retrieved 12 29, 2015, from Kuctso: <http://www.kuctso.com/hal-ini-menjadi-sulit-untuk-startup-kecil-atau-perusahaan-skala-kecil-untuk-menemukan-programmer/>
- Marisa, F. (2017). *Web Programming (Client Side and Server Side)*. Yogyakarta: CV. Budi Utama.
- riza, f. f. (2008, mei 26). Retrieved juni 10, 2015, from core.ac.uk: <http://core.ac.uk/download/pdf/11724585.pdf>

- Rouse, M. (2014). Retrieved Februari 24, 2016, from <http://searchunifiedcommunications.techtarget.com/definition/real-time-communications>
- Sasmita, W. P. (2013). ANALISIS QUALITY OF SERVICE (QOS).
- Satria, G. O. (2014). IMPLEMENTASI PROTOKOL MQTT PADA SMART BUILDING BERBASIS OPENMTC.
- Suwito, M. (2006). *Resep Makanan Jamur dari Chef Ternama*. Jakarta: PT. Agromedia Pustaka.
- Thingsboard.io. (2017). Retrieved from <https://thingsboard.io/>
- tokoteknologi.co.id. (2018). Retrieved from tokoteknologi.co.id: <http://tokoteknologi.co.id/modul-dht22-digital-temperature-and-humidity-sensor-am2302>
- Vermullard, J. (2014). M2M, IoT, DEVICE MANAGEMENT," ONE PROTOCOL TO RULE THEM.
- Widyaman, T. (2016). Retrieved from [www.warriornux.com](http://www.warriornux.com): <http://www.warriornux.com/pengertian-modul-wifi-esp8266/>
- Zhangling, Y. d. (2012). A Real-Time Group Communication Architecture Based on.

